

PENGEMBANGAN APLIKASI *E-LEARNING* DENGAN MENERAPKAN METODE *GAMIFICATION*

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Irwan Suprianto
NIM: 135150207111006



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
TAHUN
2018

PENGESAHAN

PENGEMBANGAN APLIKASI *E-LEARNING* DENGAN MENERAPKAN
METODE *GAMIFICATION*

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Irwan Suprianto
NIM: 135150207111006


Skripsi ini telah diuji dan dinyatakan lulus pada
26 Desember 2018
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I



Fajar Pradana, S.ST, M.Eng
NIP: 19871121 2015041004

Dosen Pembimbing II



Dr.Eng. Fitra Abdurrahman Bachtiar, S.T, M.Eng
NIK: 201201 8406281001

Mengetahui

Ketua Jurusan Teknik Informatika



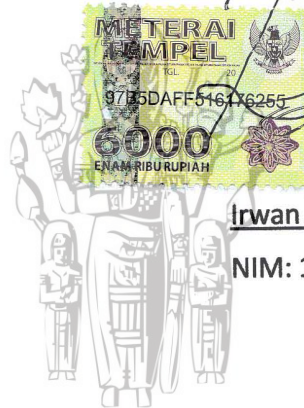
Tri Astoto Kurniawan, S.T, M.T, Ph.D
NIP: 19710518 2003121001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 10 Desember 2018



Irwan Suprianto

NIM: 135150207111006

PRAKATA

Puji syukur kehadiran Allah SWT yang telah memberikan rahmat dan karunia- Nya sehingga penulis dapat menyelesaikan skripsi dengan judul “Pengembangan Aplikasi *E-learning* Dengan Menerapkan Metode *Gamification*”. Skripsi ini disusun untuk memenuhi persyaratan untuk mendapatkan gelar Sarjana Komputer pada Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya Malang.

Melalui kesempatan ini penulis ingin menyampaikan rasa hormat dan terima kasih penulis sebesar-besarnya kepada semua pihak yang telah memberikan bantuan dan dukungan selama penulisan skripsi ini. Penulis ingin menyampaikan rasa hormat dan terima kasih kepada:

1. Kedua orang tua saya Hasnah dan Sofyan yang selalu mendukung dan selalu mendoakan penulis agar skripsi penulis dapat berjalan dengan lancar dan dimudahkan oleh Allah.
2. Bapak Fajar Pradana, S.ST, M.Eng dan Bapak Dr.Eng. Fitra Aburrachman Bachtiar, S.T, M.Eng selaku dosen pembimbing yang telah dengan sabar membimbing dan mengarahkan penulis sehingga skripsi ini dapat selesai.
3. Segenap Bapak dan Ibu Dosen Program Studi Teknik Informatika yang telah membagikan ilmunya kepada penulis.
4. Adik penulis Hairul Ramadhani Saputra yang telah mendukung dan selalu mendoakan penulis.
5. Sahabat saya tercinta Winda Hartika Sari, Thomas Ama Boleng, Herwin Yurianda dan Misbahul Munir yang telah memberikan semangat mengerjakan skripsi ini.
6. Teman penulis Muhammad Reyhan dan Reza Syahputra yang telah banyak membantu penulis dalam pembuatan skripsi ini.
7. Teman-teman dari tim basket KTM yang sudah memberikan banyak semangat dalam mengerjakan skripsi ini.
8. Seluruh pihak yang tak dapat penulis sebutkan satu per satu yang telah membantu kelancaran penyelesaian penelitian yang penulis lakukan.

Menyadari bahwa penulisan skripsi ini masih banyak kekurangan, oleh karena itu kritik dan saran yang bersifat membangun sangat diharapkan untuk menyempurnakan skripsi ini. Penulis berharap skripsi ini dapat bermanfaat bagi semua pihak.

Malang, 10 Desember 2018

Penulis

irwaansupriantoo@gmail.com

ABSTRAK

Irwan Suprianto, Pengembangan Aplikasi *E-learning* Dengan Menerapkan Metode *Gamification*

Pembimbing: Fajar Pradana, S.ST, M.Eng. dan Dr.Eng. Fitra Abdurrachman Bachtiar, S.T, M.Eng

Media pembelajaran *online* atau sering kali disebut *e-learning* adalah sebuah media yang sangat populer digunakan untuk dunia pendidikan di era sekarang ini. Tidak adanya ketertarikan siswa dengan sebuah media yang digunakan oleh sekolah menjadi penyebab utama tidak efektifnya sebuah *e-learning*. Ketertarikan atau kepuasan siswa tersebut dapat dipengaruhi oleh beberapa faktor seperti fitur atau konten yang ada pada sebuah *e-learning* tidak dapat membantu siswa untuk menyelesaikan masalah mereka, sehingga sering kali siswa tidak kembali menggunakan aplikasi tersebut. Untuk mengatasi permasalahan tersebut dibangun aplikasi *e-learning* dengan menerapkan metode *game* pada implementasi fitur-fitur atau kontennya. Metode tersebut disebut *gamification*, metode *gamification* memiliki tujuan utama yaitu meningkatkan ketertarikan. Aplikasi *e-learning* dibangun dalam bentuk web, karena web merupakan media yang dapat diakses dimana saja dan kapan saja. Sistem aplikasi *e-learning* dengan menerapkan metode *gamification* yang dibangun telah berhasil diuji dengan menggunakan metode *whitebox testing* untuk pengujian unit dan *blackbox testing* untuk pengujian *requirement*.

Kata kunci: *aplikasi e-learning, metode gamification, waterfall model, pengembangan berorientasi objek.*

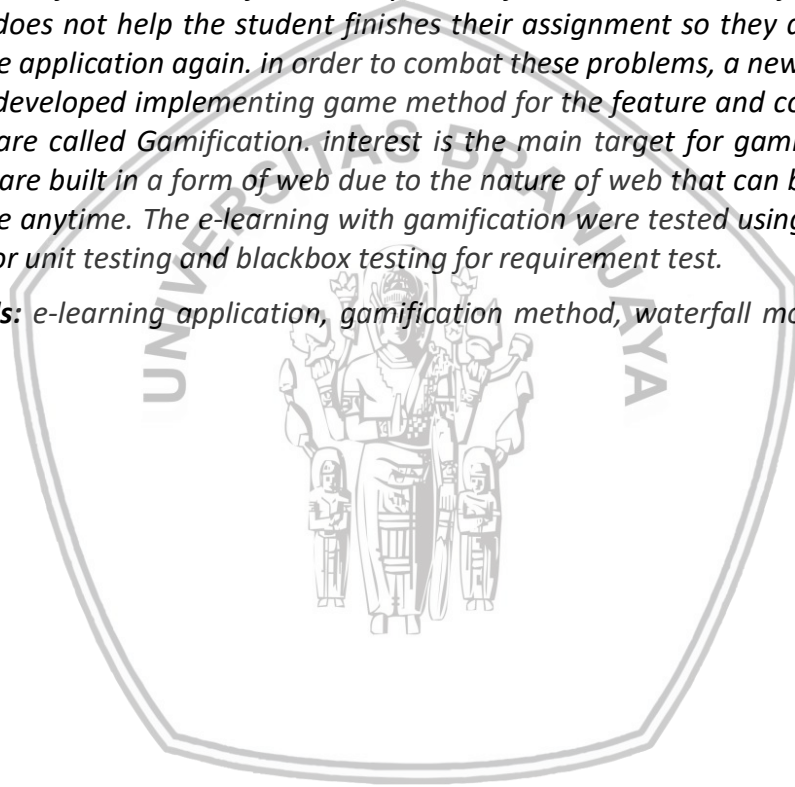
ABSTRACT

Irwan Suprianto, Pengembangan Aplikasi *E-learning* Dengan Menerapkan Metode *Gamification*

Supervisors: Fajar Pradana, S.ST, M.Eng. and Dr.Eng. Fitra Abdurrachman Bachtiar, S.T, M.Eng.

Online learning platform, also known as e-learning is a very popular media for education these days. the lack of interest from the students to the media used by the school are the main cause for the ineffective of the e-learning. interest and student satisfaction are influenced by some factor such as the feature and content does not help the student finishes their assignment so they decided not to use the application again. in order to combat these problems, a new e-learning app are developed implementing game method for the feature and content. This method are called Gamification. interest is the main target for gamification. e-learning are built in a form of web due to the nature of web that can be accessed anywhere anytime. The e-learning with gamification were tested using white box testing for unit testing and blackbox testing for requirement test.

Keywords: *e-learning application, gamification method, waterfall model, object oriented.*



DAFTAR ISI

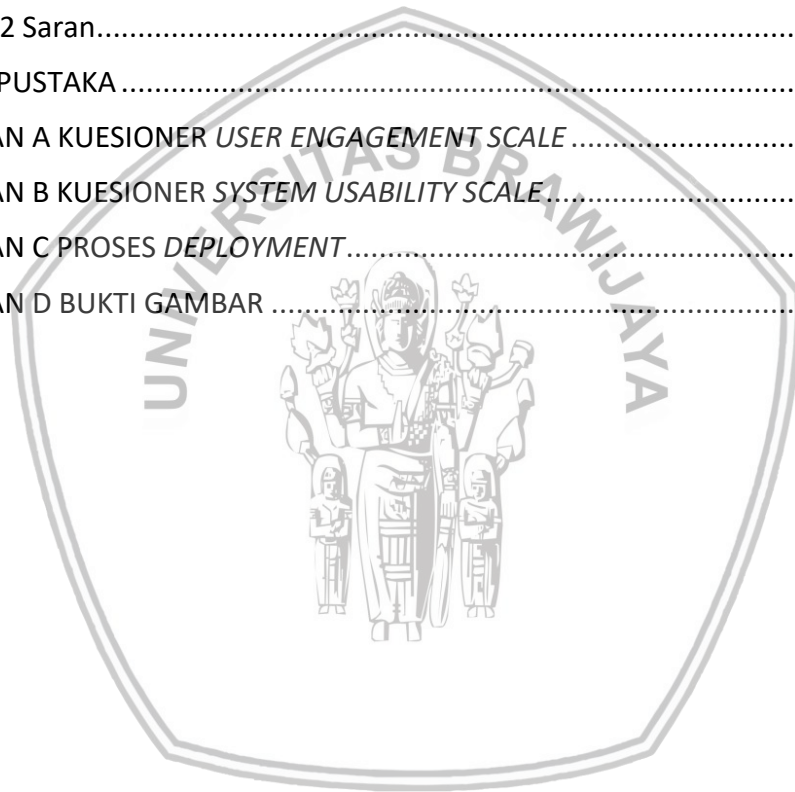
PENGESAHAN	Error! Bookmark not defined.
PERNYATAAN ORISINALITAS	Error! Bookmark not defined.
PRAKATA	iii
ABSTRAK.....	v
ABSTRACT	vi
DAFTAR ISI	vii
DAFTAR TABEL	xii
DAFTAR GAMBAR	xv
DAFTAR LAMPIRAN.....	xvii
BAB 1 PENDAHULUAN	1
1.1 Latar belakang	1
1.2 Rumusan masalah	3
1.3 Tujuan.....	3
1.4 Manfaat	3
1.5 Batasan masalah	3
1.6 Sistematika pembahasan	4
BAB 2 LANDASAN KEPUSTAKAAN	6
2.1 Kajian Pustaka.....	6
2.2 <i>E-Learning</i>	7
2.3 Rekayasa Perangkat Lunak	7
2.3.2 Proses Rekayasa Perangkat Lunak.....	8
2.4 <i>User Engagement Scale</i>	8
2.5 <i>System Usability Scale</i>	10
2.6 Model <i>Waterfall</i>	11
2.7 <i>Gamification</i>	12
2.7.1 Komponen Gamifikasi.....	12
2.8 Codeigniter	14
2.8.1 <i>Model-View-Controller</i>	14
2.8.2 <i>Model</i>	14
2.8.3 <i>View</i>	15

2.8.4 Controller	15
2.9 Unified Modeling Language	15
2.9.1 Usecase Scenario	16
2.9.2 Usecase Diagram	16
2.9.3 Class Diagram	17
2.9.4 Sequence Diagram	17
2.10 Pengujian	17
2.10.1 WhiteBox Testing.....	18
2.10.2 BlackBox Testing.....	19
BAB 3 METODOLOGI	20
3.1 Studi literatur.....	21
3.2 Analisis kebutuhan sistem.....	21
3.3 Perancangan sistem	22
3.4 Implementasi sistem.....	22
3.5 Pengujian.....	23
3.6 Penutup	23
BAB 4 ANALISIS KEBUTUHAN	24
4.1 Gambaran Umum Sistem.....	24
4.2 Identifikasi Aktor.....	24
4.3 Daftar Kebutuhan Fungsional.....	25
4.4 Daftar Kebutuhan Non Fungsional	30
4.5 Pemodelan Kebutuhan.....	31
4.5.1 Use Case Diagram.....	31
4.5.2 Use Case Scenario.....	33
BAB 5 PERANCANGAN SISTEM	50
5.1 Arsitektur Gamification dalam E-learning.....	50
5.2 Pemodelan Sequence diagram	50
5.2.1 Login.....	51
5.2.2 Membuat Kelas.....	51
5.2.3 membuat soal.....	52
5.2.4 Melakukan competition	53
5.2.5 Melihat Profile Siswa	54

5.3 Pemodelan <i>Class Diagram</i>	55
5.4 Perancangan Basis Data	56
5.5 Perancangan Komponen	56
5.5.1 <i>Controller C_Guru</i>	57
5.5.2 <i>Model M_Guru</i>	58
5.5.3 <i>Model M_Index</i>	58
5.6 Perancangan Antarmuka.....	59
5.6.1 Halaman Depan	59
5.6.2 Halaman Depan siswa.....	60
5.6.3 Halaman Depan Guru	61
5.6.4 Halaman <i>Competition</i>	62
5.6.5 Halaman Soal <i>Drag & Drop</i>	63
BAB 6 Implementasi.....	65
6.1 Implementasi <i>Database</i>	65
6.2 Implementasi Algoritme.....	65
6.2.1 <i>Controller C_Guru</i>	65
6.2.2 <i>Model M_Guru</i>	67
6.2.3 <i>Model M_Index</i>	67
6.3 Implementasi Antarmuka.....	67
6.3.1 Halaman depan Aplikasi.....	67
6.3.2 Halaman <i>Login</i>	68
6.3.3 Halaman Depan Siswa	68
6.3.4 Halaman Depan Guru	69
6.3.5 Halaman Buat Soal.....	69
6.3.6 Halaman Kelas	70
6.3.7 Halaman Aktifitas Siswa.....	70
6.3.8 Halaman Soal.....	71
6.3.9 Halaman <i>Competition</i>	71
6.3.10 Halaman Peringkat	72
BAB 7 pengujian dan analisis	73
7.1 Pengujian <i>Whitebox</i>	73
7.1.1 Pengujian Unit Klas <i>C_Index</i>	73

7.1.2 Pengujian Unit Klas C_Guru	77
7.1.3 Pengujian Unit Klas C_Siswa	80
7.2 Pengujian <i>Blackbox</i>	83
7.2.1 Pengujian <i>requirement login</i>	83
7.2.2 Pengujian <i>requirement register untuk siswa</i>	84
7.2.3 Pengujian <i>requirement register untuk guru</i>	85
7.2.4 Pengujian <i>requirement</i> membuat soal	85
7.2.5 Pengujian <i>requirement</i> membuat kelas	86
7.2.6 Pengujian <i>requirement</i> melihat daftar siswa (guru)	87
7.2.7 Pengujian <i>requirement</i> melihat <i>profile</i> siswa	87
7.2.8 Pengujian <i>requirement upload</i> materi	87
7.2.9 Pengujian <i>requirement</i> menghapus soal	88
7.2.10 Pengujian <i>requirement</i> memperbarui soal	88
7.2.11 Pengujian <i>requirement</i> melihat daftar soal	89
7.2.12 Pengujian <i>requirement upload</i> tugas (guru)	89
7.2.13 Pengujian <i>requirement</i> keluar	90
7.2.14 Pengujian <i>requirement</i> melihat daftar guru	90
7.2.15 Pengujian <i>requirement</i> melihat daftar siswa(admin)	90
7.2.16 Pengujian <i>requirement</i> melihat daftar aktifitas	91
7.2.17 Pengujian <i>requirement</i> menghapus akun guru	91
7.2.18 Pengujian <i>requirement</i> menghapus akun siswa	91
7.2.19 Pengujian <i>requirement</i> melakukan <i>competition</i>	92
7.2.20 Pengujian <i>requirement</i> melakukan <i>challenge</i>	92
7.2.21 Pengujian <i>requirement</i> melihat peringkat	93
7.2.22 Pengujian <i>requirement</i> melihat <i>profile</i>	94
7.2.23 Pengujian <i>requirement upload</i> tugas (siswa)	95
7.2.24 Pengujian <i>requirement</i> tambah kelas	95
7.2.25 Pengujian <i>requirement</i> menurunkan poin	96
7.2.26 Pengujian <i>requirement</i> melakukan <i>quiz</i>	97
7.2.27 Pengujian <i>requirement</i> melihat <i>about</i>	98
7.2.28 Pengujian <i>requirement</i> melihat <i>contact</i>	98
7.3 User Engagement Scale	98

7.4 <i>System Usability Scale</i>	100
7.5 Analisis.....	102
7.5.1 Analisis pengujian unit.....	102
7.5.2 Analisis pengujian <i>requirement</i>	102
7.5.3 Analisis <i>user engagement scale</i>	102
7.5.4 Analisis <i>system usability scale</i>	103
BAB 8 PENUTUP	104
8.1 Kesimpulan	104
8.2 Saran.....	105
DAFTAR PUSTAKA	106
LAMPIRAN A KUESIONER <i>USER ENGAGEMENT SCALE</i>	109
LAMPIRAN B KUESIONER <i>SYSTEM USABILITY SCALE</i>	115
LAMPIRAN C PROSES <i>DEPLOYMENT</i>	118
LAMPIRAN D BUKTI GAMBAR	119

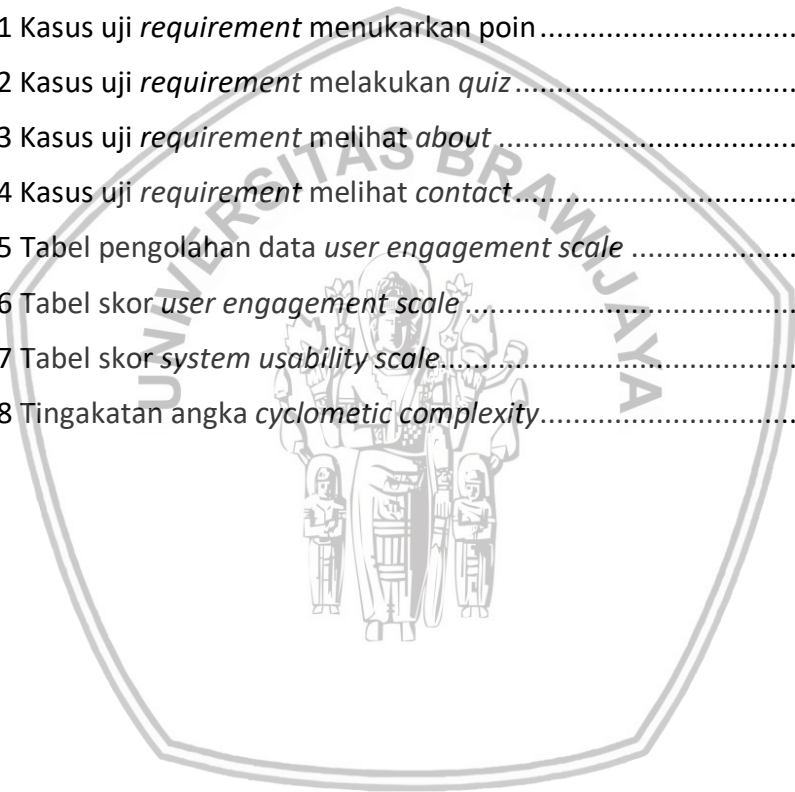


DAFTAR TABEL

Tabel 2.1 Penelitian terkait	6
Tabel 2.2 Komponen gamifikasi	13
Tabel 2.3 Tingkatan angka <i>cyclometric complexity</i>	19
Tabel 4.1 Tabel identifikasi aktor	24
Tabel 4.2 Tabel spesifikasi kebutuhan fungsional sistem.....	25
Tabel 4.3 Tabel spesifikasi kebutuhan non fungsional	31
Tabel 4.4 <i>Use case scenario</i> melakukan <i>login</i>	33
Tabel 4.5 <i>Use case scenario</i> melakukan daftar	33
Tabel 4.6 <i>Use case scenario</i> membuat soal	34
Tabel 4.7 <i>Use case scenario</i> membuat kelas	35
Tabel 4.8 <i>Use case scenario</i> melihat daftar siswa.....	36
Tabel 4.9 <i>Use case scenario</i> melihat <i>profile</i> siswa	36
Tabel 4.10 <i>Use case scenario</i> upload materi.....	37
Tabel 4.11 <i>Use case scenario</i> menghapus soal	37
Tabel 4.12 <i>Use case scenario</i> memperbarui soal	38
Tabel 4.13 <i>Use case scenario</i> melihat daftar soal	38
Tabel 4.14 <i>Use case scenario</i> upload tugas	39
Tabel 4.15 <i>Use case scenario</i> keluar	39
Tabel 4.16 <i>Use case scenario</i> melihat daftar guru	40
Tabel 4.17 <i>Use case scenario</i> melihat daftar siswa	40
Tabel 4.18 <i>Use case scenario</i> melihat daftar aktifitas	41
Tabel 4.19 <i>Use case scenario</i> menghapus akun guru.....	41
Tabel 4.20 <i>Use case scenario</i> menghapus akun siswa	42
Tabel 4.21 <i>Use case scenario</i> melakukan <i>competition</i>	42
Tabel 4.22 <i>Use case scenario</i> melakukan <i>challenge</i>	43
Tabel 4.23 <i>Use case scenario</i> melakukan <i>quiz</i>	44
Tabel 4.24 <i>Use case scenario</i> melihat peringkat.....	45
Tabel 4.25 <i>Use case scenario</i> melihat <i>profile</i>	45
Tabel 4.26 <i>Use case scenario</i> upload tugas	46
Tabel 4.27 <i>Use case scenario</i> download materi.....	46

Tabel 4.28 <i>Use case scenario</i> memilih kelas	47
Tabel 4.29 <i>Use case scenario</i> tukar poin	48
Tabel 4.30 <i>Use case scenario</i> melihat <i>about</i>	48
Tabel 4.31 <i>Use case scenario</i> melihat <i>contact</i>	49
Tabel 5.1 Algoritme fungsi tambah_kelas	57
Tabel 5.2 Algoritme fungsi tambah_kelas	58
Tabel 5.3 Algoritme fungsi create_log.....	58
Tabel 6.1 Algoritme fungsi tambah_kelas	66
Tabel 6.2 Algoritme fungsi tambah_kelas	67
Tabel 6.3 Algoritme Fungsi create_log	67
Tabel 7.1 Algoritme login.....	73
Tabel 7.2 Kasus uji algoritme login.....	76
Tabel 7.3 Algoritme tambah_kelas.....	77
Tabel 7.4 Kasus uji algoritme tambah_tugas	79
Tabel 7.5 Algoritme tukar_poin	80
Tabel 7.6 Kasus uji algoritme tukar poin	82
Tabel 7.7 Kasus uji <i>requirement login</i>	83
Tabel 7.8 Kasus uji <i>requirement register</i> untuk siswa	84
Tabel 7.9 Kasus uji <i>requirement register</i> untuk guru	85
Tabel 7.10 Kasus uji <i>requirement</i> membuat soal	85
Tabel 7.11 Kasus uji <i>requirement</i> membuat kelas	86
Tabel 7.12 Kasus uji <i>requirement</i> melihat daftar siswa (guru)	87
Tabel 7.13 Kasus uji <i>requirement</i> melihat <i>profile</i> siswa	87
Tabel 7.14 Kasus uji <i>requirement upload</i> materi	87
Tabel 7.15 Kasus uji <i>requirement</i> menghapus soal.....	88
Tabel 7.16 Kasus uji <i>requirement</i> memperbarui soal.....	88
Tabel 7.17 Kasus uji <i>requirement</i> melihat daftar soal.....	89
Tabel 7.18 Kasus uji <i>requirement upload</i> tugas (guru)	89
Tabel 7.19 Kasus uji <i>requirement</i> keluar.....	90
Tabel 7.20 Kasus uji <i>requirement</i> melihat daftar guru	90
Tabel 7.21 Kasus uji <i>requirement</i> melihat daftar siswa (admin).....	90
Tabel 7.22 Kasus uji <i>requirement</i> melihat daftar aktifitas.....	91

Tabel 7.23 Kasus uji <i>requirement</i> menghapus akun guru.....	91
Tabel 7.24 Kasus uji <i>requirement</i> menghapus akun siswa	91
Tabel 7.25 Kasus uji <i>requirement</i> melakukan <i>competition</i>	92
Tabel 7.26 Kasus uji <i>requirement</i> melakukan <i>challenge</i>	92
Tabel 7.27 Kasus uji <i>requirement</i> melihat peringkat.....	93
Tabel 7.28 Kasus uji <i>requirement</i> melihat <i>profile</i>	94
Tabel 7.29 Kasus uji <i>requirement upload</i> tugas (siswa).....	95
Tabel 7.30 Kasus uji <i>requirement</i> tambah kelas	95
Tabel 7.31 Kasus uji <i>requirement</i> menukarkan poin	96
Tabel 7.32 Kasus uji <i>requirement</i> melakukan <i>quiz</i>	97
Tabel 7.33 Kasus uji <i>requirement</i> melihat <i>about</i>	98
Tabel 7.34 Kasus uji <i>requirement</i> melihat <i>contact</i>	98
Tabel 7.35 Tabel pengolahan data <i>user engagement scale</i>	98
Tabel 7.36 Tabel skor <i>user engagement scale</i>	100
Tabel 7.37 Tabel skor <i>system usability scale</i>	101
Tabel 7.38 Tingkatan angka <i>cyclometric complexity</i>	102



DAFTAR GAMBAR

Gambar 2.1 <i>System usability scale</i> skor.....	11
Gambar 2.2 Model waterfall (Pressman, 2010).	11
Gambar 2.3 <i>Model-view-controller</i> (Griffiths, 2010).....	14
Gambar 2.4 <i>Flowchart</i> dan <i>flow graph</i>	18
Gambar 3.1 Diagram blok metode pelaksanaan.....	20
Gambar 4.1 <i>Use case diagram</i>	32
Gambar 5.1 Arsitektur <i>E-learning</i> Dengan <i>Gamification</i>	50
Gambar 5.2 <i>Sequence diagram</i> login	51
Gambar 5.3 <i>Sequence diagram</i> membuat kelas	52
Gambar 5.4 <i>sequence diagram</i> membuat Soal.....	53
Gambar 5.5 <i>Seuquence diagram</i> melakukan <i>competition</i>	54
Gambar 5.6 <i>Sequence diagram</i> melihat <i>profile</i> siswa	54
Gambar 5.7 <i>Class diagram</i>	55
Gambar 5.8 <i>Entity relationship diagram</i> (ERD).....	56
Gambar 5.9 Halaman depan	59
Gambar 5.10 Halaman depan siswa.....	60
Gambar 5.11 Halaman depan guru	61
Gambar 5.12 Halaman <i>competition</i>	62
Gambar 5.13 Halaman soal <i>drag & drop</i>	63
Gambar 6.1 Implementasi <i>database</i>	65
Gambar 6.2 Halaman depan aplikasi.....	68
Gambar 6.3 Halaman <i>login</i>	68
Gambar 6.4 Halaman depan siswa	69
Gambar 6.5 Halaman depan guru	69
Gambar 6.6 Halaman buat soal.....	70
Gambar 6.7 Halaman kelas	70
Gambar 6.8 Halaman aktifitas siswa	71
Gambar 6.9 Halaman soal	71
Gambar 6.10 Halaman <i>competition</i>	72
Gambar 6.11 Halaman peringkat	72

Gambar 7.1 Flow graph algoritme login	75
Gambar 7.2 Flow graph algoritme tambah kelas.....	78
Gambar 7.3 Flow graph algoritme tukar_poin.....	81
Gambar 7.4 SUS skor	101



DAFTAR LAMPIRAN

LAMPIRAN A KUESIONER <i>USER ENGAGEMENT SCALE</i>	109
A.1 Responden satu	109
A.2 Responden dua	111
A.3 Responden tiga	113
LAMPIRAN B KUESIONER <i>SYSTEM USABILITY SCALE</i>	115
B.1 Responden satu	115
B.2 Responden dua	116
B.3 Responden tiga	117
LAMPIRAN C PROSES <i>DEPLOYMENT</i>	118
LAMPIRAN D BUKTI GAMBAR	119
D.1 Gambar	119



BAB 1 PENDAHULUAN

1.1 Latar belakang

Dunia pendidikan di era perkembangan teknologi yang sangat maju harus didukung oleh media pembelajaran yang dapat membantu proses pembelajaran baik didalam sekolah maupun diluar sekolah. Media pembelajaran yang dimaksud adalah sebuah media pembelajaran berbasis *online* atau sebuah *e-learning*. *E-learning* adalah proses pembelajaran berbasis elektronik yang menggunakan jaringan komputer sebagai medianya atau sebuah representasi dari pembelajaran dengan menggunakan media elektronik (F. Pradana, 2018). Dengan dikembangkannya di jaringan komputer memungkinkan untuk dikembangkan dalam bentuk berbasis *web*, sehingga kemudian dikembangkan ke jaringan komputer yang lebih luas yaitu internet. Penyajian *e-learning* berbasis *web* ini bisa menjadi lebih interaktif. Dengan menggunakan *e-learning* pengguna dapat melakukan proses pembelajaran dimanapun dan kapanpun (M. Urh, 2015). Menurut Paul Jen-Hwa Hu dan Wendy Hui *E-learning* memiliki kelebihan yaitu, jangkauan geografis yang luas, nyaman dan kemudahan dalam mengakses informasi, dan biaya penerapan yang efektif dan efisien dalam penerapan teknologi pendukung (Hu, P. J. H., & Hui, W, 2012).

Saat ini *e-learning* tidak hanya digunakan pada pendidikan tinggi yang berbasis Pendidikan Jarak Jauh (PJJ), namun juga dapat diterapkan sebagai pendukung *blended learning* (F.A. Bachtiar, 2014). Permasalahan utama dalam penerapan sebuah sistem *e-learning* saat ini adalah tingkat kepuasan yang sangat rendah terhadap *e-learning*. Menurut Fajar Pradana dalam penelitiannya yang mengutip beberapa peneliti lain seperti (Lewis, C., 2002) , (Arbaugh, 2002), (Chen, 2003) telah menemukan bahwa ada enam faktor yang berdampak pada kepuasan: siswa, guru, teknologi, *course*, perancangan sistem dan faktor lingkungan (F. Pradana, 2018). Sedangkan kepuasan pengguna adalah faktor yang paling berpengaruh terhadap implementasi sebuah sistem (Delon, 1992). Tidak hanya tingkat kepuasan yang menjadi permasalahan utama dalam membuat sebuah media pembelajaran berbasis *online* atau *e-learning*, tidak adanya motivasi yang dapat mendorong siswa atau peserta didik dalam menggunakan *e-learning* karena penggunaan teknik yang salah dalam memotivasi peserta didik. Menurut Fajar Pradana terdapat beberapa alasan mengapa user memilih berhenti menggunakan pembelajaran online salah satunya penggunaan teknik yang salah dalam memotivasi peserta didik. Motivasi siswa menjadi penting dalam *e-learning* mengingat tidak adanya pengawasan dari pengajar (F. Pradana, 2018). Bentuk soal yang di aplikasikan dalam sistem *e-learning* juga salah satu faktor yang bisa mengurangi motivasi belajar siswa, dengan bentuk soal yang bervariasi dapat membuat *e-learning* ini menjadi semakin interaktif seperti yang dikatakan oleh (E. Priyambodo, 2012) dalam penelitiannya mengutip (Turkoguz, 2012) bahwa pembelajaran dengan perangkat media visual memberikan manfaat, seperti menggali kejujuran emosional individu, menciptakan komunikasi yang efektif dan menyediakan fleksibilitas dalam aktivitas belajar.

Untuk meningkatkan efisiensi, efektifitas, keminatan, motivasi siswa dalam belajar dan juga menggunakan sebuah sistem pembelajaran berbasis *online* atau *e-learning* dapat digunakan sebuah metode yaitu *gamification*. *Gamification* adalah penggunaan elemen terkait dengan video *game* (*game mechanics* dan *game dynamics*) dalam aplikasi *non-game*. Hal ini bertujuan untuk meningkatkan keterlibatan orang dan untuk mempromosikan perilaku tertentu (Simões, 2013). dimana elemen permainan digunakan untuk meningkatkan motivasi, keterlibatan dan kinerja para siswa. Aspek *game* yang diterapkan antara lain adalah poin. Setiap siswa akan diberikan poin apabila mampu menyelesaikan soal dan tantangan yang diberikan (F. Pradana, 2018). *Gamification* telah diterapkan di berbagai domain beberapa tahun terakhir, Salah satu domain tersebut adalah pendidikan dan pelatihan (Delon, 1992). Dengan menerapkan *e-learning* yang dipadukan dengan *gamification* akan membantu lingkungan belajar yang dapat membuat peserta didik merasa lebih dimotivasi, tertarik, dan juga terpancing untuk menyelesaikan tugas yang diberikan (A. Amriani, 2013). Sebelumnya sudah dilakukan penelitian oleh (K.M. Kapp, 2012) dengan judul "*The Gamification of Learning and Instruction*" namun hal ini hanya sebatas kerangka kerja yang belum diterapkan secara nyata menggunakan sebuah sistem pembelajaran berbasis *online* atau *e-learning*. Sama halnya dengan penelitian yang dilakukan (F. L. Khaleel, 2016) dengan judul "*Gamification Elements For Learning Application*" yang menjelaskan pengaruh elemen-elemen *gamification* yang diterapkan pada peserta didik, namun pada penelitian tersebut belum ada sistem yang digunakan seperti media pembelajaran *online* atau sebuah *e-learning*.

Dengan adanya beberapa permasalahan diatas penulis membuat sebuah media pembelajaran berbasis *online* atau sebuah *e-learning* yang menerapkan *gamification* dalam penelitian yang berjudul "*Pengembangan Aplikasi E-Learning Dengan menerapkan metode gamification*", dimana nantinya aplikasi *e-learning* tersebut akan diterapkan pada sekolah menengah atas atau SMA. Model pengembangan yang akan digunakan adalah *waterfall model*. *Waterfall model* yang biasa disebut *classic life cycle*, menunjukkan sebuah pendekatan yang sistematis, sekuensial untuk pengembangan perangkat lunak yang dimulai dengan spesifikasi kebutuhan konsumen atau user dan dikembangkan melalui perencanaan, pemodelan, implementasi, penyebaran (Pressman, 2010) . Untuk melakukan analisis kepuasan dan ketertarikan siswa terhadap sistem yang dibuat akan digunakan dua metode yaitu *User engagement scale* (UES), dan *System usability scale* (SUS). Menurut (Brooke, 1986) SUS adalah skala sederhana, dimana 10 skala item akan memberikan pandangan global tentang penilaian yang subjektif terhadap antarmuka yang ada. SUS nantinya akan ditampilkan dalam bentuk kuesioner, dimana tujuan utama dari kuesioner tersebut adalah untuk mencari nilai kepuasan *user* terhadap sistem. Sedangkan untuk UES metode yang digunakan untuk mencari nilai ketertarikan *user* ketika sedang berinteraksi dengan sistem atau sedang menggunakan sistem. Menurut (Heather L. O'Brien, 2018) UES adalah kualitas pengalaman pengguna yang

dicirikan oleh kenyamanan aktor ketika berinteraksi dengan sistem digital.

1.2 Rumusan masalah

Dengan adanya deskripsi dari latar belakang didapatkan beberapa rumusan masalah untuk penelitian ini, dibawah ini adalah tiga rumusan masalah yang sudah ditentukan :

1. Bagaimana hasil analisis ketertarikan dan kepuasan siswa terhadap aplikasi *e-learning* dengan menerapkan metode *gamification* ?
2. Bagaimana perancangan dan implementasi aplikasi *e-learning* dengan menggunakan metode *gamification* ?
3. Bagaimana hasil pengujian aplikasi *e-learning* menggunakan *BlackBox Testing* dan *WhiteBox Testing* ?

1.3 Tujuan

Dibawah ini adalah beberapa tujuan dari skripsi yang sudah ditentukan oleh penulis:

1. Meningkatkan ketertarikan dan kepuasan belajar siswa dengan menggunakan sistem pembelajaran *online* atau aplikasi *e-learning*.
2. Mengimplementasikan aplikasi *e-learning* yang sudah dirancang dengan menerapkan metode *gamification*.
3. Melakukan pengujian pada sistem *e-learning* dengan menggunakan *WhiteBox Testing* dan *BlackBox Testing*.

1.4 Manfaat

Dengan adanya aplikasi pembelajaran yang diberi nama *e-learning* berbasis *web* ini, diharapkan dapat meningkatkan kegiatan belajar mengajar pada sekolah menengah atas maupun sekolah menengah kejuruan.

1.5 Batasan masalah

Dalam penelitian ini memiliki ruang lingkup yang dibatasi, yaitu fokus terhadap tahap-tahap rekayasa perangkat lunak untuk membangun aplikasi *e-learning* dengan menerapkan metode *gamification*, sehingga dibutuhkan batasan masalah yang sudah dirumuskan sebagai berikut :

1. Sistem ini akan dibangun dengan menggunakan *platform* berbasis *web* dan difungsikan oleh para siswa serta tenaga pengajar sekolah menengah atas atau sekolah menengah kejuruan.
2. Sistem ini dikembangkan dengan menggunakan *framework codeigniter* dan Bahasa pemrograman PHP(*Hypertext Preprocessor*).
3. Sistem ini menggunakan MySQL sebagai database penyimpanan data. HTML, CSS serta javascript digunakan agar menambah baik dari segi tampilan.

1.6 Sistematika pembahasan

Dibawah ini adalah sistematika untuk menyusun naskah laporan penelitian ini sebagai berikut :

BAB I PENDAHULUAN

Bab ini adalah bab pertama yang akan menjelaskan beberapa bagian seperti latar belakang yang berisi kumpulan-kumpulan masalah yang timbul dan metode untuk menyelesaikan masalah tersebut, setelah latar belakang terdapat rumusan masalah yang berisi pertanyaan-pertanyaan terkait dengan permasalahan yang ada pada latar belakang, beberapa tujuan kenapa penilitian ini dilakuan, serta manfaat dilakukannya penilitian ini, kemudian ada batasan masalah agar tidak meluasnya pembahasan dalam penilitian ini, dan yang terkahir adalah sistematika pembahasan yang berisi bagian-bagian yang ada dalam dokumen penilitian ini.

BAB II LANDASAN KEPUSTAKAAN

Merupakan dasar teori yang akan digunakan untuk mendasari hubungan dengan penelitian yang akan dilakukan. Dan nantinya akan digunakan untuk menyelesaikan beberapa masalah yang ada pada latarbelakang.

BAB III METEDOLOGI PENELITIAN

Merupakan bab yang menjelaskan metode-metode penelitian yang akan digunakan. metode yang digunakan pada penelitian ini seperti studi literatur, analisis sistem, sampai dengan tahap akhir yaitu menentukan kesimpulan dan saran.

BAB IV ANALISIS KEBUTUHAN

Dalam analisis kebutuhan akan dijelaskan beberapa bagian seperti kebutuhan fungsional, non-fungsional, dan juga gambaran sistem sampai dengan identifikasi aktor.

BAB V PERANCANGAN SISTEM

Pada bagian ini akan berisi diagram-diagram yang menjelaskan tentang aplikasi *e-learning* yang dibangun dengan menerapkan metode *gamification* salah satu diagramnya adalah *sequence diagram* yang akan menjelaskan langkah-langkah untuk mencapai tujuan.

BAB VI IMPLEMENTASI

Merupakan bagian yang akan menjelaskan hasil implementasi yang sudah berhasil dilakukan dari

implemntasi basis data sampai dengan implementasi antarmuka.

BAB VII PENGUJIAN DAN ANALISIS

Bab pengujian menjelaskan secara detail pengujian yang akan digunakan dalam melakukan pengujian berdasarkan parameter-parameter yang sudah ditetapkan, dan setelah dilakukan pengujian akan dilakukan analisis terhadap hasil pengujian yang sudah dilakukan sebelumnya.

BAB VIII PENUTUP

Bab ini menjelaskan isi dari hasil penarikan kesimpulan berdasarkan hasil pengujian dan analisis pada bab sebelumnya, dan berisi saran-saran yang bisa digunakan untuk melakukan penelitian yang sama dengan pengembangan selanjutnya.



BAB 2 LANDASAN KEPUSTAKAAN

Landasan kepustakaan menjelaskan beberapa teori yang digunakan dan metode yang berkaitan dengan penelitian yang dibuat. Beberapa landasan pengetahuan yang digunakan pada penelitian ini seperti *e-learning*, rekayasa perangkat lunak(RPL), *gamification*, komponen *gamification*, model *waterfall*, *system usability scale*(SUS), *user engagement scale*(UES), *unified modeling language*(UML) yang terdiri dari diagram *usecase* diagram, diagram *class*, diagram *sequence* dan diagram *activity*, dan juga pengujian yang terdiri dari pengujian *blackbox testing*, *whitebox testing*.

2.1 Kajian Pustaka

Pada Tabel 2.1 adalah penelitian-penelitian yang menggunakan metode *gamification*. (A. Amriani, 2013) menggunakan metode *gamification* dalam penelitiannya yang digabungkan dengan *e-learning*, dalam jurnalnya disebutkan bahwa penelitian yang dilakukan lebih berfokus kepada pengaruh *gamification* terhadap lingkungan belajar. Sedangkan pada penelitian yang dilakukan oleh (F. L. Khaleel, 2016) hanya fokus pada metode *gamification* yang tidak menggunakan media *e-learning*. Pada penelitian yang dilakukan (Nurmala, 2017) yaitu membangun sebuah *e-learning* yang diberi nama *E-school* dan tidak menggunakan metode *gamification* didalamnya.

Tabel 2.1 Penelitian terkait

Penulis (Tahun)	Metode	Keterangan
(F. L. Khaleel, 2016)	<i>Gamification</i>	Elemen-elemen dari <i>gamification</i> yang bisa diterapkan dalam sebuah media pemberlajaran berbasis <i>online</i> atau <i>e-learning</i> .
(A. Amriani, 2013)	<i>Gamification</i>	Dampak penggunaan metode <i>gamification</i> terhadap lingkungan media pembelajaran berbasis <i>online</i> atau <i>e-learning</i>
(Nurmala, 2017)	<i>Evolutionary Prototype</i>	Membangun aplikasi <i>e-school</i> atau <i>e-learning</i> menggunakan pendekatan <i>evolutionary prototype</i> tanpa menerapkan metode <i>gamification</i>

2.2 E-Learning

E-learning adalah proses pembelajaran berbasis elektronik yang menggunakan jaringan komputer sebagai medianya atau sebuah representasi dari pembelajaran dengan menggunakan media elektronik (F. Pradana, 2018). Dengan dikembangkannya di jaringan komputer memungkinkan untuk dikembangkan dalam bentuk berbasis *web*, sehingga kemudian dikembangkan ke jaringan komputer yang lebih luas yaitu internet. Penyajian *e-learning* berbasis *web* ini bisa menjadi lebih interaktif (F. Pradana, 2018). Menurut Paul Jen-Hwa Hu dan Wendy Hui *E-learning* memiliki kelebihan yaitu, jangkauan geografis yang luas, nyaman dan kemudahan dalam mengakses informasi, dan biaya penerapan yang efektif dan efisien dalam penerapan teknologi pendukung (Hu, P. J. H., & Hui, W, 2012).

Dengan adanya sebuah *e-learning* dapat mempermudah guru maupun siswa dalam proses belajar mengajar baik diwaktu sekolah ataupun tidak diwaktu sekolah dan juga *e-learning* bisa diakses dimanapun. Kegiatan belajar yang konvensional bisa didukung dengan adanya sebuah *e-learning* seperti pengajar memberikan ujian dan tugas secara *online*. Tidak hanya disisi pengajar, adanya *e-learning* juga dapat memudahkan siswa seperti mendapatkan materi dengan mudah dengan mengunduh materi. Dalam jurnalnya (F. Pradana, 2018) menyebutkan bahwa kekurangan *e-learning* adalah kurangnya interaksi antara peserta didik dan pengajar atau bahkan antar peserta didik itu sendiri. Minimnya pengawasan dari pengajar menjadikan sebagian peserta kurang terlibat pada pembelajaran dengan *e-learning* ini.

2.3 Rekayasa Perangkat Lunak

Rekayasa perangkat lunak(RPL) adalah sebuah ciptaan (*product*) yang dibuat dan dikembangkan oleh seorang yang memiliki profesi pada bidang ini yang dalam waktu yang lama. Perangkat lunak atau sering disebut juga PL yang dimaksud didalamnya adalah program yang dijalankan didalam komputer dengan berbagai ukuran dan struktur. Dan akan menampilkan sebuah isi ketika program-program tersebut dijalankan, terdapat informasi-informasi yang bersifat deskriptif(menjelaskan) dalam bentuk salinan tercetak dan dalam bentuk berkas-berkas salinan secara virtual yang mencakup hampir semua media elektronik.

Rekayasa perangkat lunak meliputi suatu proses didalamnya, metode-metode dan sederatan alat-alat yang membantu para professional mengembangkan perangkat lunak komputer yang memiliki kualitas yang sangat tinggi (Pressman, 2010). Menurut Ian sommerville perangkat lunak adalah suatu bidang keilmuan yang membahas mengenai seluruh aspek-aspek yang terkait pembangunan perangkat lunak yang dimulai dari tahapan identifikasi serta pemeliharaan yang merupakan tahap akhir (Sommerville, 2003). Berikut alasan mengapa rekayasa perangkat lunak dikatakan sebagai hal yang penting, yaitu :

1. Harus dapat membuat sistem yang handal serta terpercaya secara ekonomis dan cepat karena *user* individual serta *user* organisasi bergantung pada sistem perangkat lunak yang canggih.
2. Penggunaan rekayasa perangkat lunak biasanya lebih murah. Pada sistem kebanyakan biasanya nilai *cost* akan bertambah terus seiring dengan penggunaannya.

Dua poin diatas tersebut menjelaskan bahwa sangatlah penting untuk membangun sebuah sistem yang handal dengan nilai *cost* yang murah (Sommerville, I., 2011).

2.3.2 Proses Rekayasa Perangkat Lunak

Sebuah kerangka kerja untuk menetapkan proses perancangan perangkat lunak yang lengkap dengan cara mengidentifikasi sejumlah aktifitas yang bisa diaplikasikan kedalam semua proyek perangkat lunak, terlepas dari ukuran dan kompleksitasnya. Proses yang umum untuk rekayasa perangkat lunak mencakup lima aktivitas yaitu :

1. Komunikasi
2. Perencanaan
3. Memodelkan
4. Pembangunan
5. Penyebaran

Lima kerangka kerja yang umum ini dapat digunakan selama proyek yang dikerjakan masih dalam skala program kecil dan sederhana, untuk pembuatan aplikasi *web* yang besar, dan juga sebuah rekayasa sistem berbasis komputer yang besar. Proses perangkat lunak yang digunakan akan sangat berbeda untuk setiap kasus, tetapi aktivitas yang dilakukan akan tetap sama (Pressman, 2010).

2.4 User Engagement Scale

User engagement scale (UES) adalah kualitas pengalaman pengguna yang dicirikan oleh kedalaman investasi aktor ketika berinteraksi dengan sistem digital (Heather L. O'Brien, 2018). Tujuan utama digunakannya UES adalah untuk mengetahui nilai ketertarikan *user* setelah berinteraksi dengan sistem atau menggunakan sistem. Nilai ketertarikan tersebut didapatkan setelah *user* mengisi kuesioner yang terdiri dari beberapa pertanyaan.

Dibawah ini adalah enam dimensi yang telah didefinisikan oleh (Heather L. O'Brien, 2018):

1. *Focused attention* , adalah fokus pengguna ketika sedang berinteraksi dan kehilangan banyak waktunya (7 item).
2. *Perceived usability*, pengaruh negatif yang dialami sebagai akibat dari interaksi, tingkat kontrol, dan usaha yang dikeluarkan (8 item).
3. *Aesthetic appeal*, merupakan pendapat pengguna mengenai tampilan antarmuka sebuah sistem.

4. *Endurability*, merupakan evaluasi keseluruhan yang dirasakan, dari fitur yang disediakan oleh sistem. Sehingga pengguna dapat menyarankan ke orang lain untuk menggunakan sebuah sistem tertentu.
5. *Novelty*, merupakan rasa ingin tau pengguna terhadap konten yang disediakan oleh sistem.
6. *Felt involment*, adalah persepsi pengguna terhadap ketertarikan, keterlibatan, dan kenyamanan terhadap sebuah sistem

Dalam penelitiannya (Heather L. O'Brien, 2018) menggunakan dua metode UES yaitu *UES-long form* atau UES-LF dan juga *UES-short form* atau UES-SF. Pada penelitian ini penulis menggunakan UES-LF yang memiliki pertanyaan sebagai berikut :

1. *I lost myself in this experience.*
2. *I was so involved in this experience that I lost track of time.*
3. *I blocked out things around me when I was using E-learning.*
4. *When I was using E-learning, I lost track of the world around me.*
5. *The time I spent using E-learning X just slipped away.*
6. *I was absorbed in this experience.*
7. *During this experience I let myself go.*
8. *I felt frustrated while using this Application E-learning.*
9. *I found this E-learning confusing to use.*
10. *I felt annoyed while using E-learning .*
11. *I felt discouraged while using this E-learning.*
12. *Using this E-learning was taxing*
13. *This experience was demanding.*
14. *I felt in control while using this E-learning.*
15. *I could not do some of the things I needed to do while using E-learning.*
16. *This E-learning was attractive*
17. *This E-learning was aesthetically appealing*
18. *I liked the graphics and images of E-learning.*
19. *E-learning appealed to be visual senses.*
20. *The screen layout of E-learning was visually pleasing.*
21. *Using E-learning was worthwhile*
22. *I consider my experience a success.*
23. *This experience did not work out the way I had planned.*
24. *My experience was rewarding.*
25. *I would recommend Application X to my family and friends*
26. *I continued to use Application X out of curiosity.*
27. *The content of Application X incited my curiosity.*
28. *I was really drawn into this experience.*
29. *I felt involved in this experience.*

30. *This experience was fun.*

Pertanyaan nomor 21 sampai dengan 30 digambarkan dengan *reward factor*. *reward factor* merupakan komposisi pertanyaan bagian dari *Endurability*, *Novelty*, *Felt involment* (Heather L. O'Brien, 2018).

2.5 System Usability Scale

Menurut (Brooke, 1986) *system usability scale* (SUS) adalah skala yang sederhana, dimana 10 skala item memberikan pandangan global tentang penilaian yang subjektif terhadap antarmuka yang ada. Tujuan utama dari SUS adalah untuk mencari nilai kepuasan siswa terhadap sistem yang diterapkan, nilai kepuasan tersebut didapatkan setelah *user* mengisi kuesioner yang memiliki beberapa pertanyaan dan skala tertentu.

Skala yang dimaksud dimuat dalam sebuah kuesioner yang dapat dilihat dalam **Lampiran B**. SUS adalah sebuah *linkert scale*. (Brooke, 1986) juga mendefinisikan untuk mendapatkan nilai skor, untuk pertanyaan dengan nomor 1,3,5,7,9(nomor ganjil) adalah pertanyaan yang memiliki nilai positif, sedangkan untuk pertanyaan dengan nomor 2,4,6,8,10(nomor genap) adalah pertanyaan yang memiliki nilai buruk(*negative*). Dari semua pertanyaan tersebut direpresentasikan menggunakan *linkert scale* yang sudah disebutkan sebelumnya, skala *linkert scale* dalam kuesioner penelitian ini adalah sebagai berikut :

- a. Sangat tidak setuju dilambangkan dengan angka 1
- b. Tidak setuju dilambangkan dengan angka 2
- c. Netral dilambangkan dengan angka 3
- d. Setuju dilambangkan dengan angka 4
- e. Sangat setuju dilambangkan dengan angka 5

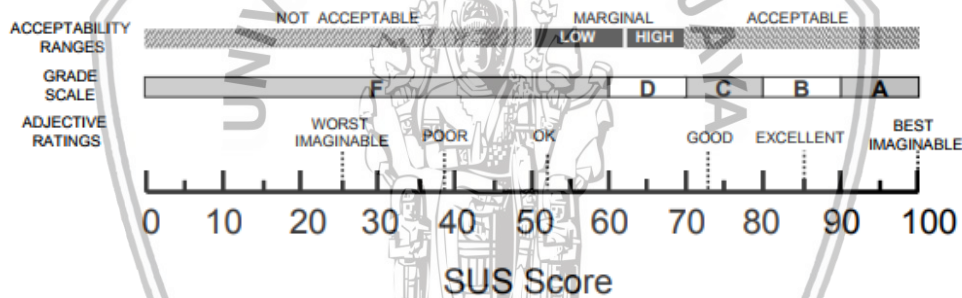
Pada pertanyaan yang bernilai positif dihitung dengan cara bobot dari setiap pertanyaan dikurangi dengan satu, sedangkan untuk pertanyaan yang bernilai negatif cara menghitungnya adalah 5 dikurangkan dengan bobot dari setiap nilai pertanyaan. Total skor bisa didapatkan dengan menjumlahkan seluruh hasil nilai negatif dan juga positif, sedangkan untuk mendapatkan nilai atau skor dari SUS yaitu dengan mengkalikan total skor dengan 2.5. Skor SUS yang didapatkan untuk setiap responden akan memiliki rentan nilai 0 sampai dengan 100.

Berikut ini merupakan 10 pertanyaan yang sudah didefinisikan oleh (Brooke, 1986) yaitu :

1. *I think that I would like to use this system frequently.*
2. *I found the system unnecessarily complex.*
3. *I thought the system was easy to use.*

4. *I think that I would need the support of a technical person to be able to use this system.*
5. *I found the various functions in this system were well integrated.*
6. *I thought there was too much inconsistency in this system.*
7. *I would imagine that most people would learn to use this system very quickly.*
8. *I found the system very cumbersome to use.*
9. *I felt very confident using the system.*
10. *I needed to learn a lot of things before I could get going with this system.*

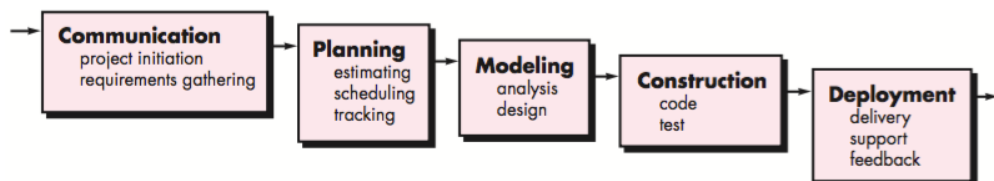
Dari skor SUS yang sudah diterima sebelumnya akan didapatkan nilai *usability* dan *acceptability* dari sebuah desain yang dikembangkan. Dalam Gambar 2.1 yang memberikan nilai range untuk skor SUS yang sudah didapatkan tadi. Untuk penilaian skor SUS yaitu berdasarkan tiga kategori yaitu *Not Acceptable* dengan rentang skor 0 sampai dengan 50.9, *Marginal* 51 sampai dengan 70.9, dan *Acceptable* 71 sampai dengan 100 (Ardiansyah, 2016).



Gambar 2.1 System usability scale skor

2.6 Model Waterfall

Model atau *Software Development Life Cycle* yang digunakan untuk membangun sistem ini adalah *waterfall*, sehingga dapat memudahkan jalannya pembangunan sistem karena kebutuhan sudah disetujui diawal. *Model waterfall* bisa digambarkan seperti dalam Gambar 2.2.



Gambar 2.2 Model waterfall (Pressman, 2010).

Menurut pressman *model waterfall* adalah model klasik yang bersifat sistematis, berurutan dalam membangun sebuah software (Pressman, 2010).

Permasalahan yang sering timbul ketika metode ini diterapkan adalah sebagai berikut (Pressman, 2010) :

1. *Project* yang nyata sangat jarang mengikuti arus tujuan dari model tersebut, akibatnya perubahan dapat menyebabkan kebingungan saat tim akan melanjutkan *project* tersebut.
2. Kebanyakan pelanggan atau konsumen kesulitan dalam menyampaikan kebutuhan yang diinginkan.
3. Pelanggan harus bersabar, karena versi program yang dibuat tidak akan keluar sampai dengan waktu yang sudah ditentukan.

2.7 Gamification

Gamification adalah penggunaan elemen terkait dengan video *game* (*game mechanics* dan *game dynamics*) dalam aplikasi *non-game*. Hal ini bertujuan untuk meningkatkan keterlibatan orang dan untuk mempromosikan perilaku tertentu (Simões, 2013). Selain metode *game base-learning*, yaitu dengan cara menggunakan *game thinking* dan elemen-elemen permainan dalam sebuah pembelajaran adalah menerapkan metode *gamification*. *Gamification* memiliki potensi besar untuk memotifasi siswa dan membuat belajar jadi lebih menarik (Lee, 2011).

Pendekatan edukasi menggunakan *gamification* memiliki keuntungan untuk memperkenalkan apa yang benar-benar penting dari dunia video game. Meningkatkan tingkat keterlibatan siswa tanpa menggunakan permainan khusus apapun. Tujuannya adalah untuk mengekstrak elemen permainan yang membuat permainan yang menyenangkan dan menyenangkan untuk dimainkan, menyesuakannya dan menggunakan elemen-elemen tersebut dalam proses pengajaran. Dengan demikian, siswa belajar, bukan dengan bermain game tertentu tetapi mereka belajar seolah-olah mereka sedang bermain game. Belajar bisa menyenangkan jika siswa belajar seolah-olah mereka sedang bermain game (Simões, 2013).

Tujuan utama dengan penerapan metode *gamification* adalah untuk meningkatkan ketertarikan siswa, memotivasi siswa, dan paham menggunakan lingkungan belajar yang nyaman dan menyenangkan, dengan demikian menghasilkan pelajar yang memiliki kualitas tinggi (F. L. Khaleel, 2016)

2.7.1 Komponen Gamification

Gagasan menerapkan konsep *gamification* pada sistem *e-Learning* sangat bergantung pada elemen perancangan permainan yang digunakan. Unsur-unsur ini memiliki peran penting untuk menciptakan *e-learning* yang efektif sekaligus untuk meningkatkan motivasi siswa selama proses berlangsung. Ada lima elemen *gamification* yang dipilih untuk diterapkan pada sistem *e-Learning*. Elemen tersebut adalah *leaderboard*, *score*, *title*, *badge*, and *completion track*, ditampilkan pada Tabel 2.2.

Tabel 2.2 Komponen gamifikasi

<i>Gamification elements</i>	<i>Description</i>
<i>Score</i>	Setiap siswa akan menerima skor untuk hasil kinerja tugas mereka dan berbagai aktifitas dalam sistem
<i>Badge</i>	Siswa akan diberikan tanda pengenalan (<i>badges</i>) dengan menyelesaikan beberapa tantangan yang berhubungan dengan aktifitas mereka.
<i>Leaderboard</i>	Siswa teratas akan ditampilkan didalam papan peringkat berdasarkan skor yang mereka miliki dan <i>badges</i> yang mereka punya.
<i>Title</i>	Setiap siswa akan mendapatkan <i>title</i> berdasarkan skor yang mereka punya, <i>title</i> tersebut ditampilkan berdasarkan <i>level</i> yang berbeda-beda dan akan dilekatkan pada nama mereka.
<i>Completion track</i>	Setiap siswa bisa melihat progres mereka didalam sistem, tugas apa yang mereka sudah selesaikan dan materi apa yang sudah mereka buka (baca).

Sumber : (A. Amriani, 2013)

Berdasarkan hasil penelitian (F. L. Khaleel, 2016) responden ditemukan :

1. Gamer menyarankan menggunakan *Points*, *Scoring system*, dan *Stars* untuk meningkatkan tingkat kesenangan dan hiburan.
2. Siswa menyarankan menggunakan *Badges*, *Top 10*, dan *Leaderboard* untuk meningkatkan motivasi bagi siswa untuk saling menantang.
3. Sisa elemen yang tidak disebutkan diatas seperti hasil, laporan, halaman depan, kompetensi, *waktu*, informasi profil, gambar, dan avatar, untuk meningkatkan keterampilan bermain dan bermain game.

• **Elemen-elemen Gamifikasi :**

1. *Game mechanics*

1. *Points*
2. *Level*
3. *Badges*
4. *Leaderboard*

2. *Game dynamics*

1. *Reward*
2. *Status*

3. *Achievement*
4. *Self-expression*

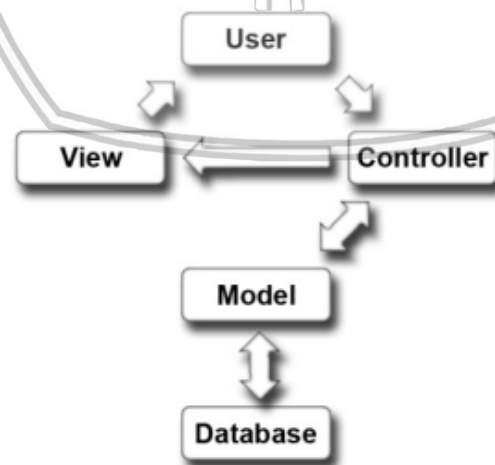
2.8 Codeigniter

Codeigniter adalah sebuah *framework* aplikasi *web* yang *open source* untuk sebuah bahasa pemrograman PHP. Codeigniter memiliki banyak fitur sehingga membuatnya unggul dari yang lainnya. Tidak seperti beberapa kerangka kerja PHP yang mungkin ditemui. Dokumentasi yang sangat lengkap mencakup setiap aspek kerangka kerja codeigniter. Codeigniter juga mampu berjalan di lingkungan *Shared Hosting* karena memiliki ukuran yang sangat kecil, namun memiliki kinerja yang sangat baik (Griffiths, 2010).

Di sisi pemrograman, Codeigniter kompatibel dengan PHP4 dan PHP5, sehingga dapat berjalan pada *host web* yang banyak digunakan pada saat ini. Codeigniter menggunakan pola desain *Model-View-Controller* (MVC), yang merupakan cara untuk mengatur aplikasi ke dalam tiga bagian yang berbeda: model-lapisan abstraksi *database*, *Views*—tampilan depan, dan *Controller*—Logika berjalannya aplikasi tersebut. Intinya, Codeigniter juga memanfaatkan secara luas pola desain singleton. Ini adalah cara untuk memuat kelas sehingga jika dipanggil beberapa kali, instance kelas yang sama akan dikembalikan. Ini sangat berguna untuk koneksi database, karena kita hanya menginginkan satu koneksi setiap kali kelas digunakan (Griffiths, 2010).

2.8.1 Model-View-Controller

MVC adalah pola desain pengembangan perangkat lunak. MVC merupakan sebuah pendekatan untuk memisahkan aplikasi menjadi tiga bagian, yaitu *Models*, *Views*, dan *Controller*. MVC membuat strukturnya dengan cara tersebut agar bisa menggunakan kode programnya kembali. (Griffiths, 2010)



Gambar 2.3 Model-view-controller

2.8.2 Model

Model adalah sebuah jenis data yang dapat digunakan oleh aplikasi, beberapa contoh data yang dapat digunakan yaitu, *Database*, *API calls*, dan

tindakan yang melibatkan pengambilan, pengembalian, memperbarui, menghapus data. (Griffiths, 2010)

2.8.3 View

Views berfungsi sebagai tempat disimpannya sebuah atau beberapa file yang bertanggung jawab untuk menampilkan data yang kita miliki kepada pengunjung halaman kita, atau pengguna untuk aplikasi kita. Tidak ada logika pemrograman, memasukkan atau memperbarui data yang berjalan pada *View*. (Blanco, 2009)

2.8.4 Controller

Controller adalah tempat jalannya logika dari sebuah aplikasi. Fungsi lain dari *controller* adalah sebagai jembatan antara *Models* dan *Views*. *Controller* akan merespon permintaan HTTP dan menghasilkan sebuah halaman *web*. (Griffiths, 2010)

2.9 Unified Modeling Language

Unified modeling language (UML) digunakan untuk pengembangan sistem perangkat lunak (PL) yang menggunakan pendekatan berorientasi objek. Intensitas penggunaan UML yang tinggi ini didukung dengan semakin matangnya konsep pemodelan yang dirumuskan dalam setiap rilis spesifikasi UML yang dikembangkan oleh Object Management Group (Kurniawan, 2018).

UML adalah bahasa atau metode permodelan utama yang digunakan untuk menganalisis, menentukan, serta merancang sistem PL (Booch, 2007).

Ada beberapa jenis diagram didalam UML menurut (Booch, 2007) sebagai berikut :

1. Structure Diagram

1. *Package diagram*
2. *Class diagram*
3. *Component diagram*
4. *Deployment diagram*
5. *Object diagram*
6. *Composite structure diagram*

2. Behavior Diagram

1. *Use case diagram*
2. *Activity diagram*
3. *State machine diagram*
4. *Interaction diagram*
 - ❖ *Sequence diagram*
 - ❖ *Communication diagram*
 - ❖ *Interaction overview diagram*
 - ❖ *Timing diagram*

2.9.1 Usecase Scenario

Usecase scenario merupakan penjelasan secara tekstual dari sekumpulan skenario interaksi. Setiap skenario mendeskripsikan urutan aksi atau langkah yang dilakukan aktor ketika berinteraksi dengan sistem, baik yang berhasil maupun gagal (Kurniawan, 2018).

Usecase scenario dijelaskan secara tekstual dalam beberapa format tergantung kebutuhannya, yaitu singkat (*brief*), informal (*casual*), atau lengkap (*fully dressed*) (Larman, 2004). yang bisa dijelaskan dalam bentuk tabel dengan satu kolom atau dua kolom (Cockburn, 2000) Pada format singkat, penjelasan diberikan cukup satu paragraf yang mengacu hanya pada skenario yang berhasil. Pada format informal, penjelasan diberikan dalam beberapa paragraf yang mencakup semua skenario, baik yang berhasil maupun gagal. Sedangkan, pada format lengkap, penjelasan dibuat secara detail disertai dengan bagian-bagian pendukung yang penting. Format terakhir ini yang banyak digunakan di dalam praktik. Bagian- bagian penting tersebut adalah (Larman, 2004) :

- **aktor primer** (*primary actor*), yaitu aktor yang menginisiasi layanan sistem untuk mencapai tujuan dari aktor tersebut. Jumlah aktor primer dimungkinkan lebih dari satu.
- **prakondisi** (*preconditions*), yaitu kondisi spesifik yang harus terpenuhi sebelum sebuah *usecase* bisa diinisiasi atau dieksekusi oleh aktor primer. Jumlah prakondisi bisa lebih dari satu keadaan.
- **alur utama** (*main or basic flow*), yaitu jalur interaksi yang mengarahkan pada skenario yang berhasil sehingga tujuan aktor bisa terpenuhi. Jalur ini hanya terdiri dari satu jalur saja.
- **alur alternatif** (*alternative flows*), yaitu jalur alternatif dari interaksi yang terjadi antara aktor dengan sistem yang mencakup pencabangan (pilihan) maupun skenario yang gagal sehingga tujuan aktor tidak terpenuhi. Jalur ini bisa terdiri dari lebih dari satu jalur kemungkinan.
- **kondisi akhir** (*postconditions*), yaitu kondisi spesifik yang harus terjadi ketika *usecase* berhasil dijalankan atau dieksekusi secara lengkap, sebagai representasi dari tujuan yang ingin dicapai oleh aktor primer. Jumlah kondisi akhir bisa lebih dari satu keadaan.

2.9.2 Usecase Diagram

Sebuah *Usecase diagram* menyatakan visualisasi interaksi yang terjadi antara pengguna (aktor) dengan sistem (Kurniawan, 2018). Diagram ini bisa menjadi gambaran yang bagus untuk menjelaskan konteks dari sebuah sistem sehingga terlihat jelas batasan dari sistem (Larman, 2004). Ada dua elemen penting yang harus digambarkan, yaitu aktor dan *Usecase*. Aktor adalah segala sesuatu yang berinteraksi langsung dengan sistem, bisa merupakan orang (yang ditunjukkan dengan perannya dan bukan namanya atau personilnya) atau sistem

komputer yang lain. Aktor dinotasikan dengan simbol gambar orang-orangan (*stick-man*) dengan nama kata benda di bagian bawah yang menyatakan peran atau sistem.

Aktor bisa bersifat primer, yaitu yang menginisiasi berjalannya sebuah *usecase*. *Usecase* dinotasikan dengan simbol elips dengan nama kata kerja aktif di bagian dalam yang menyatakan aktivitas dari perspektif aktor. Setiap aktor dimungkinkan untuk berinteraksi dengan sistem dalam banyak *usecase*. Sebaliknya, setiap *usecase* bisa dijalankan oleh lebih dari satu aktor (Kurniawan, 2018). Antar Aktor dan juga *usecase* bisa memiliki relasi, dimana sebuah *usecase* disebut sebagai *base usecase*. *Usecase* yang memiliki relasi dengan satu atau lebih *usecase* yang lainnya, disebut dengan *supplier usecase*, dalam bentuk *include* dan juga *extend* (Kurniawan, 2018). Relasi *extend* menyatakan bahwa fungsionalitas dari *base usecase* bisa diperluas oleh *supplier usecase*, jika dibutuhkan, di dalam aksekusi alur alternatif yang ada pada *usecase scenario* dari *base usecase*. Sedangkan, relasi *include* menyatakan bahwa fungsionalitas dari *base usecase* selalu hanya bisa dipenuhi dengan bantuan dari *supplier usecase* di dalam eksekusi alur utama yang ada pada *usecase scenario* dari *base usecase* (Kurniawan, 2018). Selanjutnya, sebuah aktor, disebut aktor induk, bisa memiliki relasi *inheritance* dengan aktor yang lain, disebut aktor turunan, yang menyatakan bahwa sebuah aktor merupakan turunan dari aktor yang lain. Aktor turunan akan memiliki hak akses terhadap fungsionalitas sistem yang lebih luas dibandingkan dengan aktor induk (Kurniawan, 2018).

2.9.3 Class Diagram

Class diagram merupakan diagram yang digunakan untuk mendeskripsikan objek- objek dan relasi diantara objek-objek tersebut, dan juga menunjukkan atribut- atribut dan operasi-operasi yang ada pada suatu objek (Flowler, 2003).

2.9.4 Sequence Diagram

Sequence diagram merupakan diagram yang menggambarkan operasi apa saja yang berjalan diantara objek-objek yang ada (Flowler, 2003).

2.10 Pengujian

Tahapan akhir yang dilakukan dalam suatu pengembangan perangkat lunak adalah melakukan pengujian perangkat lunak pada sistem yang selesai dibangun. Pengujian perangkat lunak merupakan hal yang sangat penting karena merupakan sebuah aktifitas untuk melakukan pemeriksaan terhadap *error* yang mungkin terjadi karena kesalahan dari perangkat lunak (Pressman, 2010).

Ada dua jenis teknik pengujian yang digunakan yaitu sebagai berikut :

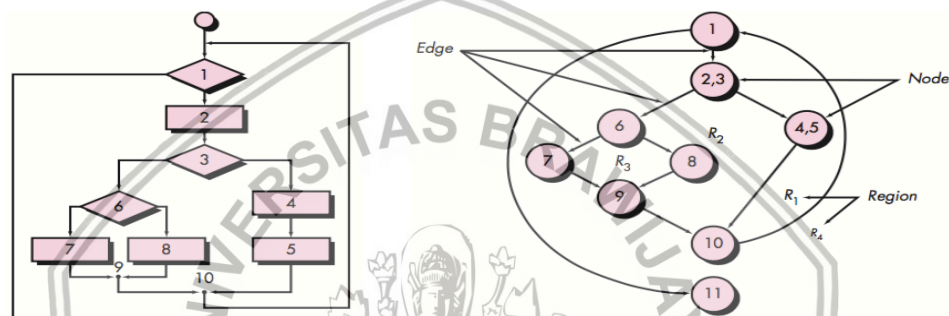
1. *Whitebox Testing*
2. *Black Box Testing*

Sedangkan untuk pengujian diluar fungsional dan juga non-fungsional sebagai berikut:

1. *User Engagement Scale*
2. *System Usability Scale*

2.10.1 WhiteBox Testing

Pengujian kotak putih adalah pengujian dengan melibatkan algoritma pengembangan sebagai bahan yang diujikan (*test case*). Dengan menggunakan *whitebox testing*, developer perangkat lunak dapat memberikan sebuah jaminan bahwa semua jalur dapat ditelusuri serta batas berhenti pada *looping*. *Basis Path Testing* merupakan salah satu *whitebox testing* yang digunakan pada penelitian. *Basis Path Testing* dimulai dengan menentukan algoritme yang selanjutnya akan dibuat menjadi notasi *flow graph* (Pressman, 2010).



Gambar 2.4 Flowchart dan flow graph

Setelah didapatkan representasi *flow graph* maka selanjutnya adalah menentukan jalur atau yang biasa disebut *independent path* dari node-node serta region yang terbentuk. Sebagai contoh jalur atau *independent path* yang terbentuk dalam Gambar 2.4 sebagai berikut :

Path 1 : 1-11

Path 2 : 1-2-3-4-5-10-11

Path 3 : 1-2-3-6-8-9-10-11

Path 4 : 1-2-3-6-7-9-10-11

Independent path telah terbentuk maka dapat dilakukan perhitungan dalam mencari jumlah jalur atau *path* dalam sebuah *flow graph*. Untuk melakukan perhitungan tersebut dikenal dengan nama *cyclomatic complexity*. *Cyclomatic complexity* merupakan sebuah perhitungan yang dapat digunakan untuk mengukur tingkatan kompleksitas dari sebuah algoritme (Pressman, 2010). Rumus *cyclomatic complexity* yang dijelaskan oleh Pressman dalam buku *software engineering* dapat dilihat pada persamaan 2-1, persamaan 2-2, dan persamaan 2-3 dibawah ini :

$$V(G) = E - N + 2 \quad (2-1)$$

$$V(G) = P + 1 \quad (2-2)$$

$$V(G) = \text{Jumlah Region} \quad (2-3)$$

Dimana :

E = Jumlah *edge* yang terdapat pada *flow graph*

N = Jumlah *node* yang terdapat pada *flow graph*

P = Jumlah *predictive node* yang terdapat pada *flow graph*

Semakin banyak jalur *independent* yang ditemukan dalam algoritme yang di uji maka, akan semakin sulit sistem untuk diujikan dan memiliki cost yang tinggi pula seperti pada pada Tabel 2.3 yang merupakan tabel angka dari *cyclometric complexity* (Nurmala, 2017).

Tabel 2.3 Tingkatan angka *cyclometric complexity*

<i>Cylometric Complexity</i>	Level Kompleksitas	Dapat Di Uji	<i>Cost dan Effort</i>
1-10	Rendah	Tinggi	Rendah
10-20	Tinggi	Sedang	Sedang
20-40	Sangat Tinggi	Rendah	Tinggi
>40	-	Tidak Dapat Di Uji	Sangat Tinggi

Sumber : (Madhavi, 2016)

2.10.2 **BlackBox Testing**

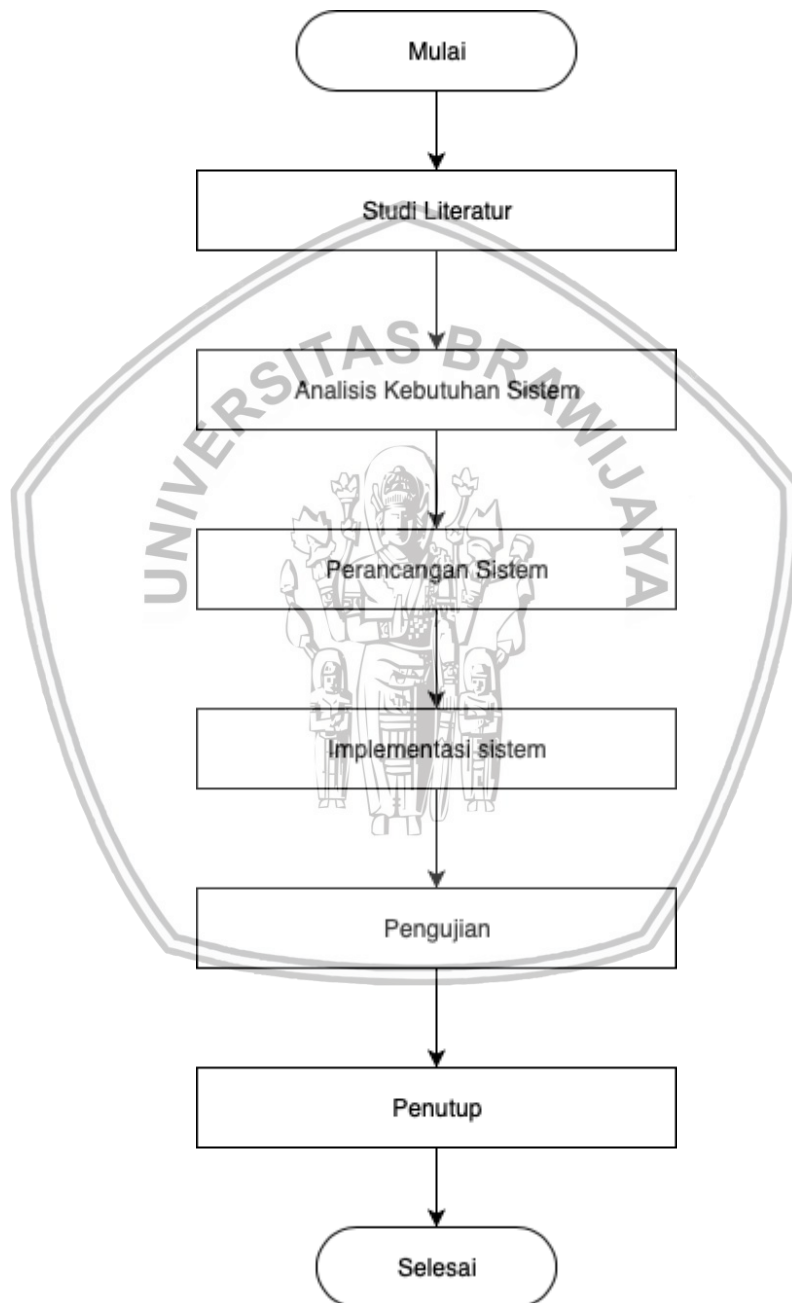
Menurut Pressman, *blackbox testing* atau yang disebut juga sebagai *behavioral testing*, pengujian ini merupakan pengujian yang berfokus kepada fungsional sistem. *Blackbox testing* melibatkan sejumlah fungsi pada sistem sebagai bahan uji, setiap fungsi diuji dengan skenario yang telah dibuat. *Blackbox testing* dilakukan sebagai pengujian untuk percobaan pencarian *error* pada sistem dengan kategori (Pressman, 2010) :

1. Fungsional sistem yang hilang serta tidak benar.
2. Terjadi kesalahan pada struktur data dan data yang diakses dalam database.
3. Adanya *error* pada antarmuka.

Blackbox testing dilakukan sebagai pelengkap dari *whitebox testing* dan bukan sebagai pengganti dari *whitebox testing*. *Blackbox testing* dan *whitebox testing* adalah salah satu paket yang tidak bisa dipisahkan. Untuk melakukan pengujian kotak hitam dibuat sebuah skenario yang terbagi kedalam kasus uji yang benar dan juga kasus uji yang salah, kemudian dari kasus tersebut dapat dilihat *feedback* yang diberikan oleh sistem apakah telah sesuai dengan kebutuhan yang dihapkan atau belum.

BAB 3 METODOLOGI

Menjelaskan mengenai tahap-tahap dalam pembangunan sistem informasi pengelolaan surat. Tahapan tersebut ada dalam Gambar 3.1 yang merupakan tahapan pengerjaan yang diilustrasikan ke dalam diagram blok.



Gambar 3.1 Diagram blok metode pelaksanaan

3.1 Studi literatur

Dalam sub studi liteatur ini akan berisi sumber dari mana teori tersebut didapatkan, dan juga memiliki tujuan sebagai dasar yang akan digunakan untuk membantu penelitian ini dapat berjalan dengan baik dari awal hingga akhir, dibawah ini ditampilkan landasan teori atau studi literature yang digunakan dan berhubungan dengan penelitian ini :

1. Rekayasa perangkat lunak
2. Model *Waterfall*
3. *Gamification*
 - a. Komponen *Gamification*
6. *Framework* CodeIgniter
7. MVC (*Model-View-Controller*)
8. *Unified Modeling Language*
 - a. *Usecase Diagram*
 - b. *Usecase Scénario*
 - c. *Sequence Diagram*
 - d. *Class Diagram*
9. Pengujian
 - a. *WhiteBox Testing*
 - b. *BlackBox Testing*
 - c. *User Engagement Scale*
 - d. *System Usability Scale*

3.2 Analisis kebutuhan sistem

Merupakan sebuah proses menemukan, menganalisis, mendefinisikan, dan mendokumentasikan kebutuhan (Hershey, 2012). Hal ini berguna untuk menggambarkan sistem secara lengkap serta menjadi perjanjian antara pengembang dan pemangku kepentingan.

Metode penggalan kebutuhan sistem dilakukan dengan diskusi bersama pemangku kepentingan dari sistem. Analisis dilakukan dengan pendekatan berorientasi objek. Setelah proses analisis kebutuhan selesai dilakukan maka akan menghasilkan:

1. Deskripsi sistem
2. Definisi kebutuhan sistem
3. Spesifikasi kebutuhan sistem
4. *Use case diagram*

5. Use case scenario

3.3 Perancangan sistem

Perancangan perlu dilakukan karena akan digunakan sebagai acuan dalam tahap implementasi. Perancangan akan dilakukan dengan pendekatan berorientasi objek. Setelah proses perancangan selesai dilakukan maka akan menghasilkan:

1. *Class diagram*
2. *Arsitektur gamification dan e-learning*
3. *Sequence diagram*
4. *Entyty Relaionship Diagram*
5. Perancangan Komponen
6. Perancangan Antar Muka

3.4 Implementasi sistem

Implementasi sistem ini akan dilakukan dengan menggunakan hasil analisis dan perancangan sebagai acuan. Implementasi sistem ini akan menghasilkan sebuah situs *web*. Situs *web* ini akan dibangun memanfaatkan berbagai bahasa dan teknologi dalam dunia komputer yaitu:

- a. Menggunakan HTML 5
- b. CSS 3 dan didukung dengan Bootstrap 3
- c. PHP 5.5 yang akan dijalankan pada *framework* Codeigniter 3
- d. Javascript yang dibuat dengan jQuery 3
- e. *Database* menggunakan MySQL

Langkah-langkah implementasi sistem sebagai berikut :

1. Mempersiapkan *framework* codeigniter, dimana codeigniter adalah aplikasi terbuka yang berupa framwork PHP dengan model MVC (*Model, View, Controller*).
2. Langkah selanjutnya adalah implementasi bootstrap yang sudah tersedia kedalam codeigniter pada folder view.
3. Setelah semua bootstrap atau tema dari sistem sudah di implementasikan serta terlihat rapi, selanjutnya menambahkan fungsi-fungsi yang ada.
4. Untuk melengkapi kinerja fungsi yang ada agar dapat berjalan sebagai mana mestinya, selanjutnya akan dibangun sebuah database dari aplikasi *e-learning*, di dalam database inilah nanti semua data akan disimpan, dari data untuk *login* , serta data yang diunggah, maupun data yang telah diperbarui.

5. Setelah semuanya telah selesai dikerjakan dari fungsi-fungsi yang ada maupun dari database yang sudah dibuat sebelumnya, langkah selanjutnya adalah melihat keseluruhan sistem apakah sistem sudah berjalan dengan baik.

3.5 Pengujian

Setelah tahap implementasi selesai dilakukan maka pengujian akan menjadi tahap selanjutnya. Pengujian perlu dilakukan untuk memastikan bahwa perangkat lunak yang dibuat telah sesuai dan dapat berfungsi dengan benar. Pada pengujian akan menghasilkan hasil uji dari :

1. *WhiteBox testing*
2. *BlackBox testing*
3. *Usability engagemen scale*
4. *System usability scale*

3.6 Penutup

Pada bagian penutup penulis akan melakukan pengambilan kesimpulan dilakukan setelah semua tahapan telah dilakukan. pada bagian penutup ini berisi dua sub bab yaitu kesimpulan dan saran.



BAB 4 ANALISIS KEBUTUHAN

Bagian ini akan menjelaskan proses menentukan kebutuhan apa saja yang dibutuhkan oleh sistem agar dapat berjalan sesuai dengan yang diharapkan oleh pengguna. Kebutuhan tersebut terbagi menjadi dua yaitu kebutuhan fungsional dan juga non fungsional yang dijelaskan dalam Tabel 4.2 dan juga dalam Tabel 4.3.

4.1 Gambaran Umum Sistem

Sistem ini akan digunakan guru, siswa, dan juga admin. Pada sistem ini guru dapat membuat soal, *upload* materi, dan juga *upload* tugas. Soal yang sudah dibuat oleh guru tadi akan dijadikan soal yang akan dikerjakan oleh siswa dalam bentuk *challenge*, quiz, dan tantangan antara siswa satu dengan yang lainnya, dan yang berhasil menyelesaikannya dengan benar akan mendapatkan sejumlah poin tertentu. Sistem ini juga dapat menampilkan susunan atau peringkat siswa yang memiliki poin tertinggi.

4.2 Identifikasi Aktor

Pada bagian ini akan menjelaskan aktor yang dapat menggunakan atau mengakses sistem, terdapat empat aktor yang sudah didefinisikan pada sistem ini seperti yang ada pada Tabel 4.1, tujuan dilakukannya identifikasi aktor adalah untuk mengetahui siapa saja yang mampu atau dapat mengakses sistem yang sudah dibuat. Proses identifikasi aktor dapat dilihat pada Tabel 4.1.

Tabel 4.1 Tabel identifikasi aktor

Aktor	Deskripsi
User	User adalah aktor pengguna yang belum teridentifikasi statusnya sehingga tidak dapat mengakses sistem secara penuh jika belum berhasil melakukan <i>login</i> . user hanya dapat menggunakan sistem untuk mendaftar dan juga melihat beberapa informasi yang disediakan sistem pada halaman depan.
Admin	Aktor ini dapat bertugas menghapus akun guru dan juga siswa serta melihat seluruh aktifitas yang terjadi didalam sistem.
Guru	Guru memiliki hubungan pewarisan (<i>inheritance</i>) dengan aktor <i>user</i> . Guru adalah aktor yang berhasil melakukan <i>login</i> dan dapat mengakses semua fitur

	yang ada dalam aplikasi <i>e-learning</i> salah satunya membuat soal.
Siswa	Siswa memiliki hubungan <i>inheritance</i> dengan <i>user</i> . Siswa adalah pengguna sistem yang dapat mengerjakan soal yang sudah dibuat oleh guru sebelumnya, soal yang dikerjakan terbagi menjadi tiga yaitu pada bagian <i>challenge</i> , <i>quiz</i> dan juga <i>competition</i> . Siswa juga dapat mengunduh materi dan juga mengunggah tugas yang diberikan guru, selain itu siswa juga dapat menukarkan poin yang sudah didapatkan dengan mengerjakan soal tadi.

4.3 Daftar Kebutuhan Fungsional

Kebutuhan fungsional dalam penelitian ini didapat dari hasil penggalan informasi dari metode *gamification* yang digunakan untuk menyelesaikan permasalahan yang ada pada latar belakang. Setiap kebutuhan yang ada telah divalidasi dan diverifikasi dengan cara melihat elemen-elemen yang ada pada metode *gamification* dan juga komunikasi dengan dosen pembimbing. Proses validasi ini menunjukkan bahwa kebutuhan yang sudah ditetapkan sesuai dengan metode yang digunakan dan telah disepakati oleh dosen pembimbing. Verifikasi menunjukkan bahwa kebutuhan aplikasi *e-learning* yang berhasil didefinisikan dapat dengan mudah dipahami. Setiap kebutuhan akan diberikan kode AEF_KFSK. Kebutuhan utama akan di kodekan dengan format KF sedangkan spesifikasi kebutuhan akan di kodekan dengan format SK pada kode kebutuhan yang sudah didefinisikan sebelumnya. Spesifikasi kebutuhan fungsional ditunjukkan pada Tabel 4.2.

Tabel 4.2 Tabel spesifikasi kebutuhan fungsional sistem

No	Spesifikasi	Kode	Aktor
1.	Sistem Harus menyediakan fungsi <i>login</i> untuk mengizinkan user mengakses sistem sebagai guru, admin, dan juga siswa.	AEF_0100	Admin, Guru, Siswa
	1.1. Aktor mengisi <i>form login</i> yang berupa <i>username</i> dan <i>password</i> .	AEF_0101	Admin, Guru, Siswa
2.	Sistem harus menyediakan fungsi keluar(<i>login</i>) agar admin, guru, dan	AEF_0200	Admin, Guru, Siswa

	siswa dapat keluar dari sistem.		
3.	Sistem harus menyediakan fungsi untuk membuat soal.	AEF_0300	Guru
	3.1. Sistem harus mampu menyediakan menu untuk memilih tipe soal, tipe soal yang dimaksud yaitu, pilihan ganda, <i>drag and Drop</i> gambar, pilihan ganda gambar dan juga acak kata.	AEF_0301	Guru
	3.2. Sistem harus menyediakan menu untuk memilih kelas dan bab mata pelajaran tertentu.	AEF_0302	Guru
	3.3. Sistem menyediakan <i>field</i> soal, jawaban, dan jawaban benar.	AEF_0303	Guru
4	Sistem harus menyediakan fungsi untuk guru membuat sebuah kelas baru.	AEF_0400	Guru
	4.1. Membuat kelas harus mengisi nama kelas, mata pelajaran, angkatan, dan <i>enroll code</i> .	AEF_0401	Guru
5.	Sistem dapat menampilkan daftar siswa setiap kelas.	AEF_0500	Guru
	5.1. Sistem menampilkan daftar siswa dalam bentuk tabel, isi tabel tersebut adalah nomer, nama, nomer hp, email, angkatan, tugas, <i>last update</i> , dan juga <i>icon</i> untuk menampilkan <i>profile</i> siswa.	AEF_0501	-
6.	Sistem dapat menampilkan <i>profile</i> siswa.	AEF_0600	Guru
	6.1. Sistem menampilkan <i>profile</i> siswa yang berisi foto <i>profile</i> , nama, alamat, nomor hp, <i>email</i> , angkatan, statistik, dan juga <i>log</i> aktifitas.	AEF_0601	-
	6.2. <i>Log</i> aktifitas menampilkan daftar aktifitas siswa dalam bentuk tabel yang berisi nomor, nama, tanggal,	AEF_0602	-

	<i>log.</i>		
	6.3. Sistem menampilkan statistik dalam bentuk tabel yang berisi nomor, matapelajaran, bab, nilai, dan juga jumlah poin.	AEF_0603	
7.	Sistem harus menyediakan fungsi <i>upload</i> materi untuk guru.	AEF_0700	Guru
	7.1. <i>Form upload</i> yang harus diisi adalah nama materi dan juga tombol <i>upload file</i> .	AEF_0701	-
	7.2. Sistem mengizinkan <i>file</i> yang dapat <i>diupload</i> berjenis .jpg, .jpeg, .png, .pdf, .doc, .docx, .xls, .xlsx, .ppt, .pptx, serta memiliki ukuran dibawah 50 mb.	AEF_0702	-
8.	Sistem harus menyediakan fungsi <i>upload</i> tugas untuk guru.	AEF_0800	Guru
	8.1. Sistem menampilkan <i>form upload</i> yang berisi nama tugas, dan juga tombol untuk memilih <i>file</i> .	AEF_0801	-
	8.2. Sistem mengizinkan <i>file</i> yang dapat <i>diupload</i> berjenis .jpg, .jpeg, .png, .pdf, .doc, .docx, .xls, .xlsx, .ppt, .pptx, serta memiliki ukuran dibawah 50 mb.	AEF_0802	-
9.	Sistem dapat menampilkan daftar guru.	AEF_0900	Admin
	9.1. Sistem menampilkan daftar guru dalam bentuk tabel yang berisi nomor, nama , alamat, nomor hp, <i>email</i> , dan juga icon untuk menghapus akun guru.	AEF_0901	-
10.	Sistem dapat menampilkan daftar siswa.	AEF_1000	Admin
	10.1. Sistem menampilkan daftar siswa dalam bentuk tabel yang berisi nomor, nama , alamat,	AEF_1001	-

	nomor hp, <i>email</i> , angkatan dan juga icon untuk menghapus akun guru.		
11.	Sistem dapat menampilkan seluruh aktifitas yang dilakukan didalam sistem baik itu aktifitas guru, admin, dan juga siswa.	AEF_1100	Admin
	11.1. Sistem menampilkan aktifitas seluruh actor dalam bentuk tabel yang berisi nomor, nama, tanggal, <i>log</i> .	AEF_1101	-
12.	Sistem harus menyediakan fungsi untuk menghapus akun guru.	AEF_1200	Admin
13.	Sistem harus menyediakan fungsi untuk menghapus akun siswa.	AEF_1300	Admin
14.	Sistem harus menyediakan fungsi <i>competition</i> .	AEF_1400	Siswa
	14.1. Sistem menampilkan <i>form competition</i> yang berisi kelas, siswa, tombol tantang, dan batal.	AEF_1401	-
	14.2. Sistem menampilkan <i>form</i> soal yang berisi waktu pengerjaan, soal, pilihan jawaban, tombol <i>next</i> dan <i>submit</i> .	AEF_1402	-
15.	Sistem harus menyediakan fungsi <i>challenge</i> untuk siswa.	AEF_1500	Siswa
	15.1. Sistem menampilkan <i>form competition</i> yang berisi kelas, bab, tombol mulai, dan batal.	AEF_1501	-
	15.2. Sistem menampilkan <i>form</i> soal yang berisi waktu pengerjaan, soal, pilihan jawaban, tombol <i>next</i> dan <i>submit</i> .	AEF_1502	
16.	Sistem harus menyediakan fungsi untuk menampilkan peringkat.	AEF_1600	Siswa

	16.1. Sistem menampilkan peringkat dalam bentuk tabel yang berisi , nomor, nama, kelas, angkatan jumlah poin.	AEF_1601	-
	16.2. Sistem mampu menyusun tampilan peringkat <i>berdasarkan</i> angkatan dan juga kelas.	AEF_1602	-
17.	Sistem harus mampu menampilkan <i>profile</i> siswa.	AEF_1700	Siswa
	17.1. Sistem menampilkan <i>profile</i> siswa yang berisi foto <i>profile</i> alamat, nomor hp, <i>email</i> , angkatan dan juga statistik.	AEF_1701	-
	17.2. Sistem menampilkan statistik dalam bentuk tabel yang berisi nomor, matapelajaran, bab, nilai, dan juga jumlah poin.	AEF_1702	-
18.	Sistem harus mampu menyediakan fungsi <i>upload tugas</i> .	AEF_1800	Siswa
	18.1. <i>Form upload</i> yang harus diisi adalah nama tugas dan juga tombol <i>upload file</i> .	AEF_1801	-
	18.2. Sistem mengizinkan <i>file</i> yang dapat diupload berjenis .jpg, .jpeg, .png, .pdf, .doc, .docx, .xls, .xlsx, .ppt, .pptx, serta memiliki ukuran dibawah 50 mb.	AEF_1802	-
19.	Sistem harus mampu menyediakan fungsi <i>download</i> materi.	AEF_1900	Siswa
20.	Sistem harus menyediakan fungsi tukar poin.	AEF_2000	Siswa
21.	Sistem harus menyediakan fungsi tambah kelas.	AEF_2100	Siswa
	21.1. Sistem menampilkan tambah kelas yang berisi piliah kelas dalam bentuk <i>dropdown</i> , juga <i>enrollment key</i> , serta tombol buat dan juga batal.	AEF_2101	-

22.	Sistem harus menyediakan fungsi daftar.	AEF_2200	Siswa, Guru
	22.1. Sistem menampilkan <i>form register</i> untuk siswa-siswa berisi <i>username, password, nama, alamat, nomor hp, email, angkatan</i> dan tombol <i>upload foto profile</i> .	AEF_2201	-
	22.2. Sistem menampilkan <i>form register</i> untuk siswa-siswa berisi <i>username, password, nama, alamat, nomor hp, email, dan tombol upload foto profile</i> .	AEF_2202	
23.	Sistem harus menyediakan fungsi untuk memperbarui soal.	AEF_2300	Guru
24.	Sistem harus menampilkan informasi menu tentang(<i>about</i>) pada halaman depan aplikasi.	AEF_2400	User
25.	Sistem harus menampilkan informasi menu kontak(<i>contatc</i>) pada halaman depan aplikasi.	AEF_2500	User
26.	System harus menyediakan fungsi untuk menampilkan daftar soal yang sudah dibuat guru.	AEF_2600	Guru
27.	Sistem harus menyediakan fungsi untuk menghapus soal.	AEF_2700	Guru
28.	Sistem harus menyediakan fungsi <i>quiz</i> .	AEF_2800	Siswa
	Sistem menampilkan <i>form quiz</i> yang berisi kelas, siswa, tombol <i>tantang</i> , dan <i>batal</i> .	AEF_2801	-
	Sistem menampilkan <i>form</i> soal yang berisi waktu pengerjaan, soal, pilihan jawaban, tombol <i>next</i> dan <i>submit</i> .	AEF_2802	-

4.4 Daftar Kebutuhan Non Fungsional

Kebutuhan non fungsional adalah kemampuan atau kualitas dari sebuah sistem. Kebutuhan non fungsional yang sudah ditetapkan berdasar hasil diskusi dengan dosen pembimbing dan berdasarkan kebutuhan sistem. Setiap

kebutuhan non fungsional dikodekan AENF_NF. Kebutuhan non fungsional yang sudah didefinisikan ditunjukkan pada Tabel 4.3.

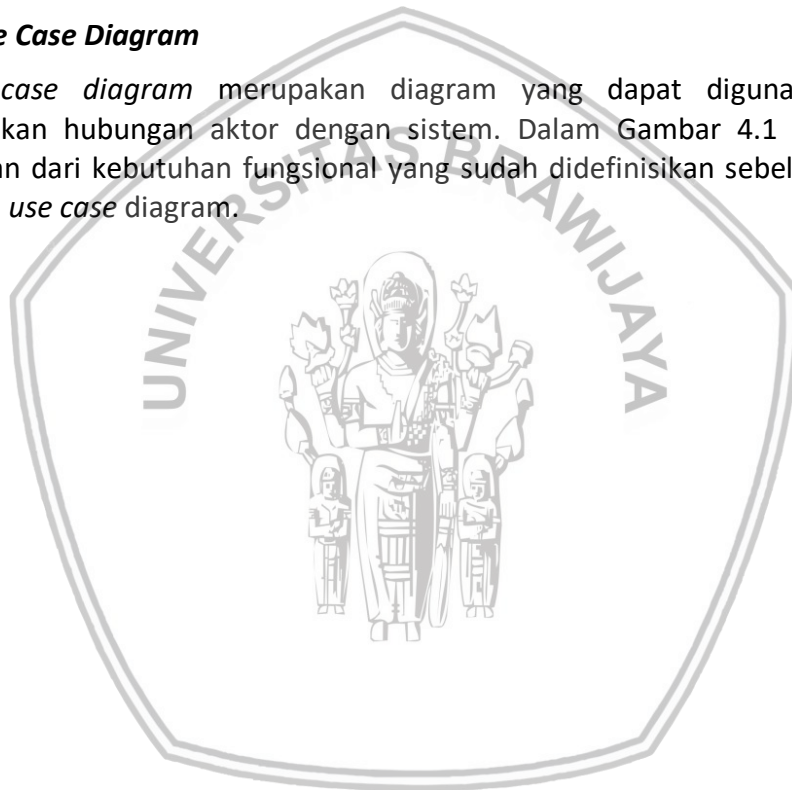
Tabel 4.3 Tabel spesifikasi kebutuhan non fungsional

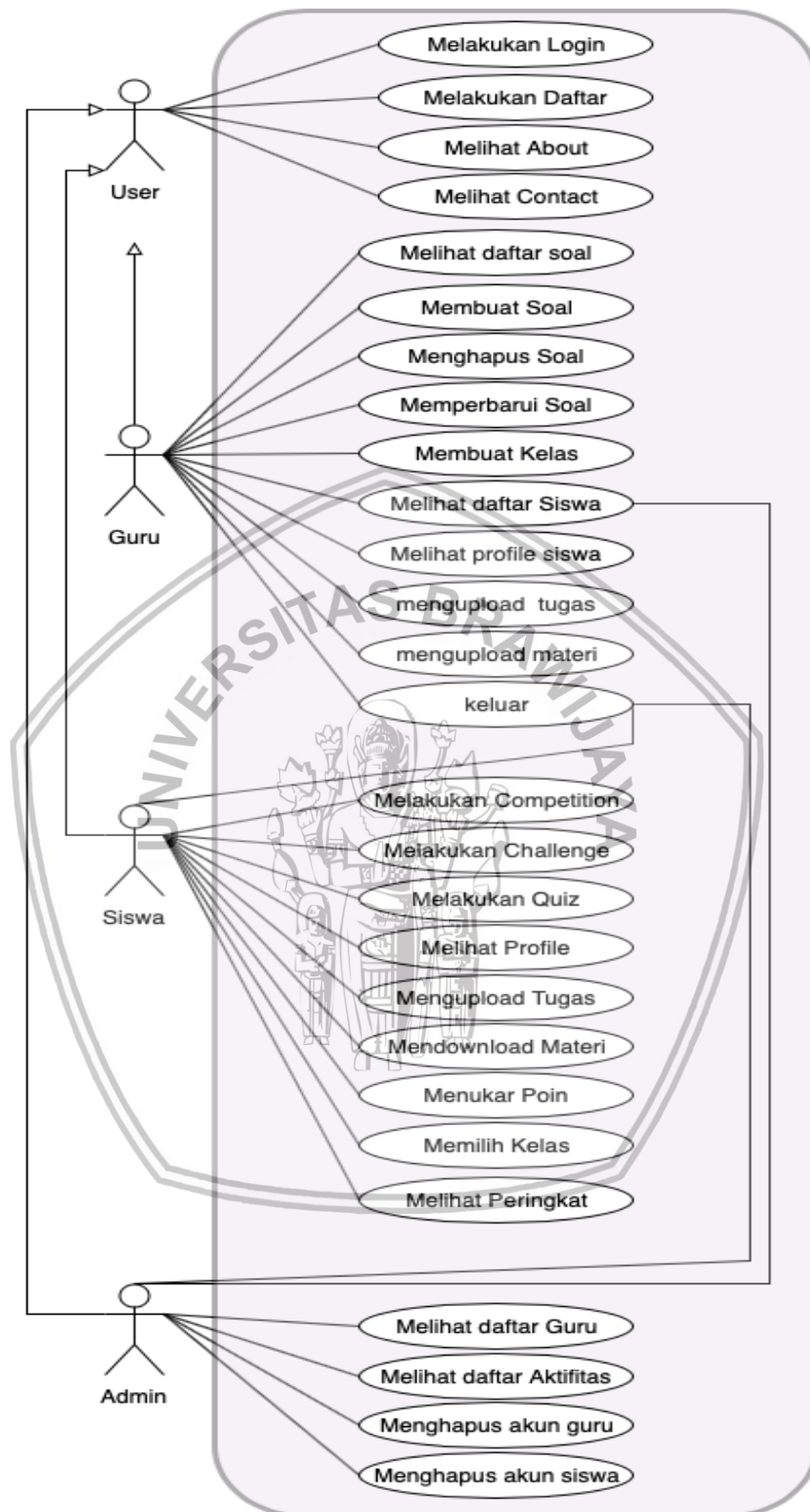
No	Parameter	Spesifikasi	Kode
1	Compability	sistem harus dapat dibuka pada <i>browser Google Chrome , Safari</i> dan juga <i>Mozila Firefox</i>	AENF_01

4.5 Pemodelan Kebutuhan

4.5.1 Use Case Diagram

Use case diagram merupakan diagram yang dapat digunakan untuk menjelaskan hubungan aktor dengan sistem. Dalam Gambar 4.1 merupakan kelanjutan dari kebutuhan fungsional yang sudah didefinisikan sebelumnya dan dijadikan *use case diagram*.





Gambar 4.1 Use case diagram

4.5.2 Use Case Scenario

Use Case scenario adalah langkah-langkah yang harus dilakukan oleh aktor untuk mencapai sebuah tujuan secara sistematis. Berdasarkan *use case* diagram yang sudah dibuat sebelumnya berhasil didapatkan dua puluh delapan scenario dan dapat dilihat dalam Tabel 4.4 sampai dengan Tabel 4.31.

4.5.2.1 Melakukan Login

Tabel 4.4 Use case scenario melakukan login

Login	
Actor	<i>User</i>
Objective	Mengizinkan <i>user</i> mengakses sistem sebagai guru, siswa, admin.
Pre-Condition	Halaman beranda aplikasi sudah terbuka
Main Flow	<ol style="list-style-type: none"> 1. User memilih tombol <i>Login</i>. 2. Sistem akan menampilkan halaman yang berisi <i>form</i> login <i>username</i> dan <i>password</i>. buat akun , lupa <i>password</i> dan juga tabel peringkat. 3. <i>User</i> memberikan masukan yaitu berupa <i>username</i> dan <i>password</i>, lalu memilih tombol login. 4. Sistem melakukan <i>validasi</i> terhadap masukan <i>User</i>.
Alternative Flow	<ol style="list-style-type: none"> 1. Jika memilih tombol <i>Login</i> tetapi <i>username</i> atau <i>password</i> terbaca kosong, maka sistem akan menampilkan pesan "<i>Please fill out this field</i>" pada kolom yang tidak di isi atau kosong. 2. <i>Username</i> dan <i>password</i> salah, sistem akan menampilkan pesan "<i>This is warning! User tidak ada</i>". 3. <i>Username</i> salah, maka sistem akan menampilkan pesan "<i>This is warning! User tidak ada</i>". 4. <i>Password</i> salah, maka sistem akan menampilkan pesan "<i>This is warning! User tidak ada</i>".
Post -Condition	<i>User</i> telah teridentifikasi sebagai admin atau guru atau siswa.

4.5.2.2 Melakukan Daftar

Tabel 4.5 Use case scenario melakukan daftar

Register

Actor	<i>User</i>
Objective	Mengizinkan <i>user</i> mendaftar kedalam sistem.
Pre-Condition	Halaman <i>Register</i> sudah terbuka
Main Flow	<ol style="list-style-type: none"> 1. User memilih tombol register yang terdiri dari dua menu (Daftar Sebagai Guru atau Daftar Sebagai Siswa). 2. Sistem akan menampilkan <i>form</i> isian sesuai dengan pilihan. 3. <i>User</i> memasukkan data <i>register</i> sebagai guru berupa <i>username</i>, <i>password</i>, nama, alamat, nomor hp, <i>email</i>, dan juga foto <i>profile</i>. Sedangkan untuk register sebagai siswa memasukkan data berupa <i>username</i>, <i>password</i>, nama, alamat, nomor hp, <i>email</i>, angkatan dan juga foto <i>profile</i>. 4. <i>User</i> memilih tombol daftar. 5. Sistem akan menyimpan data <i>User</i> kedalam <i>database</i>.
Alternative Flow	<ol style="list-style-type: none"> 1. <i>User</i> tidak memberikan <i>inputan</i> atau masukan pada salah satu atau semua <i>field</i> yang ada pada formulir <i>register</i>, sistem akan memberikan keluaran atau menampilkan pesan yang berisi "<i>Please fill out this field</i>" pada kolom yang tidak diisi. 2. <i>User</i> memilih tombol Daftar sebagai Guru, maka siswa akan menampilkan halaman daftar guru. 3. Jika <i>User</i> memilih tombol Daftar sebagai Siswa, maka system akan menampilkan halaman daftar siswa.
Post -Condition	<i>User</i> terdaftar kedalam sistem sebagai guru atau siswa.

4.5.2.3 Membuat Soal

Tabel 4.6 Use case scenario membuat soal

Membuat Soal	
Actor	Guru
Objective	Mengizinkan Guru membuat soal yang akan disimpan didalam <i>database</i>
Pre-Condition	Halaman Daftar kelas sudah terbuka
Main Flow	<ol style="list-style-type: none"> 1. Guru memilih menu buat soal. 2. Sistem akan menampilkan sub menu pilihan soal. 3. Guru memilih salah satu bentuk soal, yaitu pilihan

	<p>ganda, pilihan ganda gambar <i>drag & drop</i>, atau acak kata.</p> <p>4. Sistem akan menampilkan formulir soal sesuai dengan jenis soal yang dipilih.</p> <p>5. Guru memasukkan data untuk soal berjenis pilihan ganda berupa Kelas, Bab, Soal, Jawaban Benar.</p> <p>6. Guru memilih tombol buat soal.</p> <p>7. Sistem akan menyimpan soal kedalam basis data atau <i>database</i>.</p>
Alternative Flow	<p>1. Guru tidak memberikan <i>inputan</i> pada kolom soal, sistem akan memberikan keluaran berupa pesan "<i>Please fill out this field</i>".</p> <p>2. Guru tidak mengisi pilihan jawaban maka sistem akan menampilkan pesan "<i>Please fill out this field</i>".</p> <p>3. Guru memilih jenis soal pilihan ganda, maka sistem akan menampilkan halaman soal yang berisi <i>form</i> soal pilihan ganda.</p> <p>4. Jika Guru memilih jenis soal <i>drag & drop</i>, maka sistem akan menampilkan halaman soal yang berisi <i>form</i> soal <i>drag & drop</i>.</p> <p>5. Jika Guru memilih jenis soal pilihan acak kata, maka sistem akan menampilkan halaman soal yang berisi <i>form</i> soal acak kata.</p> <p>6. Jika guru memilih jenis soal pilihan ganda gambar, maka sistem akan menampilkan halaman soal yang berisi <i>form</i> soal pilihan ganda gambar.</p>
Post -Condition	Soal berhasil disimpan didalam database

4.5.2.4 Membuat Kelas

Tabel 4.7 *Use case scenario* membuat kelas

Membuat Kelas	
Actor	Guru
Objective	Mengizinkan Guru membuat Kelas pada sistem.
Pre-Condition	Halaman daftar kelas sudah dibuka
Main Flow	1. Guru memilih tombol tambah.

	2. Sistem akan menampilkan <i>form</i> pembuatan kelas. 3. Guru memasukkan data kelas berupa nama kelas, mata pelajaran, angkatan, keterangan kelas dan juga <i>enroll code</i> . 4. Guru memilih tombol buat. 5. Sistem menyimpan data kelas kedalam <i>database</i> .
Alternative Flow	1. Jika guru memilih tombol batal maka sistem akan menampilkan halaman daftar kelas.
Post -Condition	Guru berhasil membuat Kelas dan berada di halaman daftar kelas.

4.5.2.5 Melihat Daftar Siswa

Tabel 4.8 *Use case scenario* melihat daftar siswa

Melihat daftar Siswa	
Actor	Guru
Objective	Mengizinkan Guru melihat daftar siswa yang telah terdaftar didalam sistem.
Pre-Condition	Halaman daftar kelas sudah terbuka
Main Flow	1. Guru memilih tombol lihat siswa. 2. Sistem akan menampilkan daftar siswa sesuai dengan kelas yang dipilih dalam bentuk tabel yang berisi no, nama, nomor hp, <i>email</i> , angkatan, tugas, <i>last update</i> , dan <i>icon</i> untuk view <i>profile</i> .
Alternative Flow	-
Post -Condition	Sistem berhasil menampilkan daftar siswa.

4.5.2.6 Melihat *Profile* Siswa

Tabel 4.9 *Use case scenario* melihat *profile* siswa

Melihat <i>Profile</i> Siswa	
Actor	Guru
Objective	Mengizinkan Guru untuk melihat <i>profile</i> siswa yang telah terdaftar didalam sistem.
Pre-Condition	Halaman Daftar siswa sudah terbuka
Main Flow	1. Guru memilih <i>icon user</i> pada siswa tertentu.

	2. Sistem akan menampilkan halaman <i>profile</i> siswa yang berisi nama, foto <i>profile</i> , alamat, nomor hp, <i>email</i> , angkatan, <i>log</i> aktifitas yang berisi nama, tanggal log. Dan juga data statistik yang berisi jumlah poin, tabel dengan data mata pelajaran, bab, nilai.
Alternative Flow	-
Post -Condition	Sistem berhasil menampilkan <i>profile</i> siswa.

4.5.2.7 Upload Materi

Tabel 4.10 Use case scenario upload materi

Upload Materi	
Actor	Guru
Objective	Mengizinkan Guru untuk mengunggah materi kedalam <i>database</i> .
Pre-Condition	Halaman daftar bab sudah terbuka
Main Flow	<ol style="list-style-type: none"> 1. Guru memilih salah satu materi bab yang sudah ada. 2. Sistem akan menampilkan <i>form</i> yang berisi tugas, materi, tombol <i>upload</i> tugas dan juga tombol <i>upload</i> materi. 3. Guru memilih tombol <i>upload</i> materi. 4. sistem akan menampilkan <i>form</i> yang berisi nama materi dan juga tombol <i>choose file</i>. 5. Guru memasukkan data berupa nama materi dan memilih tombol <i>choose file</i> dan memilih <i>file</i> dari <i>device</i>. 6. Guru memilih tombol <i>upload</i>. 7. Sistem akan menyimpan materi kedalam <i>database</i>
Alternative Flow	<ol style="list-style-type: none"> 1. Jika guru tidak mengisi nama materi sistem akan memberikan <i>output</i> atau menampilkan pesan dengan isi "<i>Please fill out this field</i>" pada kolom yang tidak diisi.
Post -Condition	Sistem berhasil menyimpan data dalam <i>database</i> .

4.5.2.8 Menghapus Soal

Tabel 4.11 Use case scenario menghapus soal

Upload Tugas	
Actor	Guru

Objective	Mengizinkan Guru untuk menghapus soal yang sudah dibuat.
Pre-Condition	Halaman daftar soal sudah terbuka
Main Flow	<ol style="list-style-type: none"> 1. Guru memilih <i>icon</i> sampah pada daftar soal yang ada. 2. Sistem akan menampilkan pesan yang berisi “anda yakin?” dan juga tombol <i>cancel</i> serta ok. 3. Guru memilih tombol ok. 4. sistem akan menghapus soal yang diilih dari database.
Alternative Flow	5. Jika guru memilih tombol cancel maka soal tidak dihapus dan kembali pada halaman dafatr soal.
Post -Condition	Sistem berhasil menghapus data soal dalam <i>database</i> .

4.5.2.9 Memperbarui Soal

Tabel 4.12 *Use case scenario* memperbarui soal

Upload Tugas	
Actor	Guru
Objective	Mengizinkan Guru untuk menghapus soal yang sudah dibuat.
Pre-Condition	Halaman daftar soal sudah terbuka
Main Flow	<ol style="list-style-type: none"> 1. Guru memilih <i>icon</i> edit pada daftar soal yang ada. 2. Sistem akan menampilkan <i>form</i> soal yang berisi soal, jawaban, jawaban benar, dan tombol ubah soal. 3. Guru melakukan perubahan sesuai yang diinginkan yaitu pada soal, jawaban atau jawaban benar. 4. Guru memilih tombol ubah soal.
Alternative Flow	5. Jika guru tidak melakukan perubahan pada form soal tersebut lalu memilih tombol ubah soal, maka tidak ada perubahan pada isi dari form soal tersebut.
Post -Condition	Sistem berhasil memperbarui data soal dalam <i>database</i> .

4.5.2.10 Melihat daftar soal

Tabel 4.13 *Use case scenario* melihat daftar soal

Upload Tugas	
Actor	Guru

Objective	Mengizinkan Guru untuk melihat daftar soal.
Pre-Condition	Halaman depan guru sudah terbuka
Main Flow	1. Guru memilih menu daftar soal. 2. Sistem akan menampilkan halaman daftar soal.
Alternative Flow	
Post -Condition	Sistem berhasil menampilkan daftar soal.

4.5.2.11 Upload Tugas

Tabel 4.14 Use case scenario upload tugas

Upload Tugas	
Actor	Guru
Objective	Mengizinkan Guru untuk mengunggah tugas kedalam <i>database</i> .
Pre-Condition	Halaman daftar bab sudah terbuka
Main Flow	<ol style="list-style-type: none"> 1. Guru memilih salah satu materi bab yang sudah ada. 2. Sistem akan menampilkan <i>form</i> yang berisi tugas, materi, tombol <i>upload</i> tugas dan juga tombol <i>upload</i> materi. 3. Guru memilih tombol <i>upload</i> tugas. 4. sistem akan menampilkan <i>form</i> yang berisi nama tugas dan juga tombol <i>choose file</i>. 5. Guru memasukkan data berupa nama tugas dan memilih tombol <i>choose file</i> dan memilih <i>file</i> dari <i>device</i>. 6. Guru memilih tombol <i>upload</i>. 7. Sistem akan menyimpan materi kedalam <i>database</i>
Alternative Flow	8. Jika guru tidak mengisi nama tugas sistem akan memberikan <i>output</i> berupa pesan " <i>Please fill out this field</i> " pada kotak yang kosong.
Post -Condition	Sistem berhasil menyimpan data dalam <i>database</i> .

4.5.2.12 Keluar

Tabel 4.15 Use case scenario keluar

Upload Tugas	
Actor	Guru, Admin, Siswa

Objective	Mengizinkan Guru, Admin, Siswa keluar dari sistem.
Pre-Condition	Halaman utama sudah terbuka
Main Flow	<ol style="list-style-type: none"> 1. Guru, Admin, atau Siswa-Sisiwi memilih tombol Keluar. 2. Sistem akan menghapus <i>session login</i> dari Guru, Admin, atau Siswa 3. Sistem akan menampilkan halaman depan.
Alternative Flow	-
Post -Condition	Guru, Admin, Siswa berhasil keluar dari sistem.

4.5.2.13 Melihat Daftar Guru

Tabel 4.16 Use case scenario melihat daftar guru

Melihat daftar Guru	
Actor	Admin
Objective	Mengizinkan Admin untuk melihat Guru yang telah terdaftar didalam sistem.
Pre-Condition	Halaman <i>/log</i> aktifitas sudah ditampilkan
Main Flow	<ol style="list-style-type: none"> 1. Admin memilih menu daftar Guru. 2. Sistem menampilkan halaman dafatar Guru dalam bentuk tabel yang berisi no, nama, alamat , nomor hp, <i>email</i>, dan juga aksi.
Alternative Flow	-
Post -Condition	Sistem berhasil menampilkan halaman Daftar guru.

4.5.2.14 Melihat Daftar Siswa

Tabel 4.17 Use case scenario melihat daftar siswa

Melihat daftar Siswa	
Actor	Admin
Objective	Mengizinkan Admin untuk melihat Siswa yang telah terdaftar didalam sistem.
Pre-Condition	Halaman <i>/log</i> aktifitas sudah ditampilkan
Main Flow	<ol style="list-style-type: none"> 1. Admin memilih menu daftar Siswa. 2. Sistem menampilkan halaman daftar Siswa dalam bentuk tabel yang berisi no, nama, alamat , nomor hp,

	<i>email</i> , angkatan dan juga aksi.
Alternative Flow	-
Post -Condition	Sistem berhasil menampilkan halaman Daftar siswa.

4.5.2.15 Melihat daftar Aktifitas

Tabel 4.18 Use case scenario melihat daftar aktifitas

Melihat daftar Aktifitas	
Actor	Admin
Objective	Mengizinkan Admin untuk melihat daftar aktifitas yang terjadi didalam sistem.
Pre-Condition	Halaman depan admin sudah terbuka
Main Flow	<ol style="list-style-type: none"> 1. Admin memilih menu daftar aktifitas. 2. Sistem menampilkan halaman aktifitas dalam bentuk tabel yang berisi no, nama, tanggal , <i>log</i>.
Alternative Flow	-
Post -Condition	Sistem berhasil menampilkan halaman <i>log aktifitas</i> .

4.5.2.16 Menghapus Akun Guru

Tabel 4.19 Use case scenario menghapus akun guru

Menghapus Akun Guru	
Actor	Admin
Objective	Mengizinkan Admin untuk menghapus akun guru yang telah terdaftar kedalam sistem.
Pre-Condition	Halaman daftar Guru sudah terbuka
Main Flow	<ol style="list-style-type: none"> 1. Admin memilih tombol hapus akun pada salah satu guru yang ada ditabel. 2. Sistem akan menghapus akun guru yang dipilih dari <i>database</i>. 3. Menampilkan halaman daftar guru.
Alternative Flow	-
Post -Condition	Sistem berhasil mengahus akun guru dari <i>database</i> dan kembali ke halaman daftar guru.

4.5.2.17 Menghapus Akun Siswa

Tabel 4.20 *Use case scenario* menghapus akun siswa

Menghapus Akun Siswa	
Actor	Admin
Objective	Mengizinkan Admin untuk menghapus akun Siswa yang telah terdaftar kedalam sistem.
Pre-Condition	Halaman <i>log</i> aktifitas sudah terbuka
Main Flow	<ol style="list-style-type: none"> 1. Admin memilih tombol hapus akun pada salah satu siswa yang ada ditabel. 2. Sistem akan menghapus akun siswa yang dipilih dari <i>database</i>. 3. Menampilkan halaman daftar siswa.
Alternative Flow	-
Post -Condition	Sistem berhasil mengahus akun siswa yang dipilih dari <i>database</i> dan menampilkan halaman daftar siswa.

4.5.2.18 Melakukan *Competition*

Tabel 4.21 *Use case scenario* melakukan *competition*

Melakukan <i>Competition</i>	
Actor	Siswa
Objective	Mengizinkan Siswa menantang siswa lain untuk mengerjakan soal.
Pre-Condition	Halaman <i>Competition</i> sudah terbuka
Main Flow	<ol style="list-style-type: none"> 1. Siswa memilih kelas dan juga siswa yang ditantang. 2. Siswa memilih tombol tantang. 3. Sistem akan menampilkan pesan “anda telah berhasil menantang nama siswa, silahkan tunggu <i>response</i> selanjutnya”.
Alternative Flow	<ol style="list-style-type: none"> 1. Jika siswa menekan tombol batal maka sistem akan menampilkan halaman depan siswa.
Post -Condition	Sistem berhasil mengirimkan <i>request</i> kesiswa yang ditantang.

4.5.2.19 Melakukan Challenge

Tabel 4.22 Use case scenario melakukan challenge

Melakukan Challenge	
Actor	Siswa
Objective	Mengizinkan siswa untuk mengerjakan setiap bagian (Bab) pada mata pelajaran tertentu.
Pre-Condition	Halaman <i>Challenge</i> sudah terbuka
Main Flow	<ol style="list-style-type: none"> 1. Siswa memilih nama kelas. 2. Sistem menampilkan daftar <i>challenge</i> yang berisi nama bab, tombol mulai challenge, skor, jumlah soal. 3. Siswa memilih tombol mulai <i>challenge</i>. 4. Sistem akan menampilkan pesan "Selamat Mengerjakan Soal Challenge BAB mata pelajaran, Terdapat 10 Soal dan Anda Diberikan waktu Selama 30 Menit Untuk Mengerjakan! Anda Harus Mendapatkan Skor 80 untuk Mendapatkan Tambahan Poin! Selamat Mengerjakan!". 5. Siswa memilih tombol <i>close</i>. 6. Sistem menampilkan <i>form</i> soal yang berisi soal, jawaban, waktu, dan juga <i>progressbar</i>. 7. Siswa menjawab soal sampai dengan nomor terakhir. 8. Siswa memilih tombol submit. 9. Sistem menampilkan pesan konfirmasi. 10. Siswa memilih tombol <i>close</i>. 11. System menampilkan halaman depan siswa.
Alternative Flow	<ol style="list-style-type: none"> 12. Jika siswa berhasil mendapatkan poin lebih atau sama 80 maka sistem akan menampilkan pesan "Sistem menampilkan pesan " selamat anda berhasil mengerjakan soal dengan jumlah (jumlah soal) dan Mendapatkan Nilai : (nilai) , dan poin kamu telah mendapatkan tambhan poin". 13. Jika siswa mendapatkan poin dibawah 80 maka sistem akan menampilkan pesan " selamat anda berhasil mengerjakan soal dengan jumlah (jumlah soal) dan Mendapatkan Nilai : (nilai) , Namun Anda Tidak Mendapatkan Tambahan Poin Karena Nilai Anda Kurang Dari 80".

Post -Condition	Siswa berhasil membuka halaman <i>challenge</i> dan mengerjakan soal-soal yang ada pada bab mata pelajaran tertentu.
------------------------	--

4.5.2.20 Melakukan Quiz

Tabel 4.23 Use case scenario melakukan quiz

Melakukan Challenge	
Actor	Siswa
Objective	Mengizinkan siswa untuk mengerjakan <i>Quiz</i> yang tersedia .
Pre-Condition	Halaman <i>Quiz</i> sudah terbuka
Main Flow	<ol style="list-style-type: none"> 1. Siswa memilih tombol lihat pada kelas yang ada. 2. Sistem menampilkan daftar <i>Quiz</i>. 3. Siswa memilih tombol mulai <i>quiz</i> pada salah satu <i>Quiz</i>. 4. Sistem akan menampilkan pesan “Selamat Mengerjakan Soal Quiz BAB mata pelajaran, Terdapat 10 Soal dan Anda Diberikan waktu Selama 30 Menit Untuk Mengerjakan! Anda Harus Mendapatkan Skor 80 untuk Mendapatkan Tambahan Poin! Selamat Mengerjakan!”. 5. Siswa memilih tombol <i>close</i>. 6. Sistem menampilkan <i>form</i> soal yang berisi soal, jawaban, waktu, dan juga <i>progressbar</i>. 7. Siswa menjawab soal sampai dengan nomor terakhir. 8. Siswa memilih tombol submit. 9. Sistem menampilkan pesan konfirmasi. 10. Siswa memilih tombol <i>close</i>. 11. System menampilkan halaman depan siswa.
Alternative Flow	<ol style="list-style-type: none"> 12. Jika siswa berhasil mendapatkan poin lebih atau sama 80 maka sistem akan menampilkan pesan “Sistem menampilkan pesan “ selamat anda berhasil mengerjakan soal dengan jumlah (jumlah soal) dan Mendapatkan Nilai : (nilai) , dan poin kamu telah mendapatkan tambhan poin”. 13. Jika siswa mendapatkan poin dibawah 80 maka sistem akan menampilkan pesan “ selamat anda berhasil mengerjakan soal dengan jumlah (jumlah soal) dan Mendapatkan Nilai : (nilai) , Namun Anda Tidak Mendapatkan Tambahan Poin Karena Nilai Anda

	Kurang Dari 80”.
Post -Condition	Siswa berhasil membuka halaman <i>challenge</i> dan mengerjakan soal-soal yang ada pada bab mata pelajaran tertentu.

4.5.2.21 Melihat Peringkat

Tabel 4.24 Use case scenario melihat peringkat

Melihat Peringkat	
Actor	Siswa
Objective	Mengizinkan siswa untuk menampilkan peringkat.
Pre-Condition	Halaman depan terbuka
Main Flow	<ol style="list-style-type: none"> 1. Siswa memilih menu peringkat. 2. Sistem akan menampilkan halaman peringkat yang berisi data siswa dalam bentuk tabel dengan data, nama , kelas, angkatan , dan juga jumlah poin.
Alternative Flow	<ol style="list-style-type: none"> 1. Jika siswa memilih kelas pada menu tabel peringkat ,maka tabel akan ditampilkan berdasarkan kelas. 2. Jika siswa memilih angkatan pada menu tabel peringkat, maka tabel akan ditampilkan berdasarkan angkatan.
Post -Condition	Sistem berhasil menampilkan halaman peringkat.

4.5.2.22 Melihat Profile

Tabel 4.25 Use case scenario melihat profile

Melihat Profile	
Actor	Siswa
Objective	Mengizinkan siswa untuk menampilkan <i>Profile</i> mereka sendiri.
Pre-Condition	Halaman depan sudah ditampilkann
Main Flow	<ol style="list-style-type: none"> 1. Siswa memilih menu <i>Profile</i>. 2. Sistem akan menampilkan halaman <i>Profile</i> siswa yang berisi nama, alamat, nomor hp, <i>email</i>, angkatan, foto profile jumlah poin, dan juga data statistik yang dalam bentuk tabel dan berisi no, mata pelajaran, bab, nilai.

Alternative Flow	-
Post -Condition	Siswa berhasil membuka halaman <i>Profile</i> .

4.5.2.23 Upload Tugas

Tabel 4.26 Use case scenario upload tugas

Upload Tugas	
Actor	Siswa
Objective	Mengizinkan Siswa untuk mengunggah tugas kedalam <i>database</i> .
Pre-Condition	Halaman daftar bab sudah terbuka
Main Flow	<ol style="list-style-type: none"> 1. Siswa memilih tombol upload tugas. 2. sistem akan menampilkan <i>form</i> yang berisi nama tugas dan juga tombol <i>choose file</i>. 3. Siswa memasukkan data berupa nama tugas dan memilih tombol <i>choose file</i> dan memilih <i>file</i> dari <i>device</i>. 4. Siswa memilih tombol <i>upload</i>. 5. Sistem akan menyimpan tugas kedalam <i>database</i>.
Alternative Flow	6. Jika siswa tidak mengisi nama tugas sistem akan menampilkan tulisan " <i>Please fill out this field</i> " pada kotak yang kosong.
Post -Condition	Siswa berhasil menyimpan file kedalam sistem.

4.5.2.24 Download Materi

Tabel 4.27 Use case scenario download materi

Download Materi	
Actor	Siswa
Objective	Mengizinkan Siswa untuk mengunduh materi yang ada disistem.
Pre-Condition	Halaman daftar bab sudah terbuka
Main Flow	<ol style="list-style-type: none"> 1. Siswa memilih salah satu materi bab yang sudah ada. 2. Sistem akan menampilkan halaman detail bab yang berisi tugas, materi , tombol <i>upload</i> tugas dan

	<p>mengunduh materi.</p> <p>3. Siswa memilih tombol <i>download</i> materi.</p> <p>4. Sistem akan otomatis mengunduh <i>file</i> yang dipilih kedalam <i>device</i> Siswa.</p>
Alternative Flow	-
Post -Condition	Siswa berhasil menyimpan file kedalam <i>device</i> .

4.5.2.25 Memilih Kelas

Tabel 4.28 Use case scenario memilih kelas

Memilih Kelas	
Actor	Siswa
Objective	Mengizinkan Siswa untuk memilih kelas yang tersedia pada sistem.
Pre-Condition	Halaman tambah kelas sudah terbuka
Main Flow	<ol style="list-style-type: none"> 1. Siswa memilih kelas, dan juga memasukkan <i>enrollment</i> key. 2. Siswa memilih tombol buat. 3. Sistem menampilkan pesan konfirmasi “anda yakin ingin bergabung dengan kelas ini?”. 4. Siswa memilih tombol ok. 5. System menampilkan pesan konfirmasi berupa “selamat anda berhasil menambahkan kelas”. 6. Siswa memilih tombol <i>close</i>. 7. Sistem akan menambahkan kelas dan menampilkan halaman depan siswa.
Alternative Flow	<ol style="list-style-type: none"> 1. Jika siswa menekan tombol batal, maka sistem akan secara otomatis kembali pada halam kelas.
Post -Condition	Siswa berhasil membuat kelas.

4.5.2.26 Tukar Poin

Tabel 4.29 Use case scenario tukar poin

Tukar Poin	
Actor	Siswa
Objective	Mengizinkan Siswa untuk menukarkan poin yang mereka miliki dengan nilai pengetahuan, kognitif, sikap, dan juga keterampilan.
Pre-Condition	Halaman tukar poin sudah terbuka
Main Flow	<ol style="list-style-type: none"> 1. Siswa memilih nilai yang ingin ditukar, dan memasukkan jumlah poin sesuai dengan nilai yang ditukar. 2. Siswa menekan tombol tukar poin. 3. System menampilkan pesan konfirmasi"anda yakin ingin menukarkan poin". 4. Siswa memilih tombol ok. 5. Sistem akan secara otomatis mengurangi jumlah poin yang siswa miliki dan menampilkan halaman tukar poin.
Alternative Flow	<ol style="list-style-type: none"> 1. Jika siswa tidak memiliki poin yang cukup saat memilih tombol tukar poin maka sistem akan menampilkan pesan "this is a warning! Poin anda tidak dapat ditukarkan, karena jumlah poin anda kurang untuk penukaran ini! Terima kasih".
Post -Condition	Siswa berhasil menurunkan poin yang dimiliki dengan salah satu pilihan nilai dan kembali kehalaman tukar poin.

4.5.2.27 Melihat About

Tabel 4.30 Use case scenario melihat about

Tukar Poin	
Actor	User
Objective	Mengizinkan User untuk bisa mengakses halaman <i>about</i> .
Pre-Condition	Halaman depan aplikasi sudah terbuka
Main Flow	<ol style="list-style-type: none"> 1. User memilih menu <i>about</i>. 2. Sistem menampilkan halaman <i>about</i>.

Alternative Flow	-
Post -Condition	<i>User</i> berhasil mengakses halaman <i>about</i> .

4.5.2.28 Melihat *contact*

Tabel 4.31 Use case scenario melihat *contact*

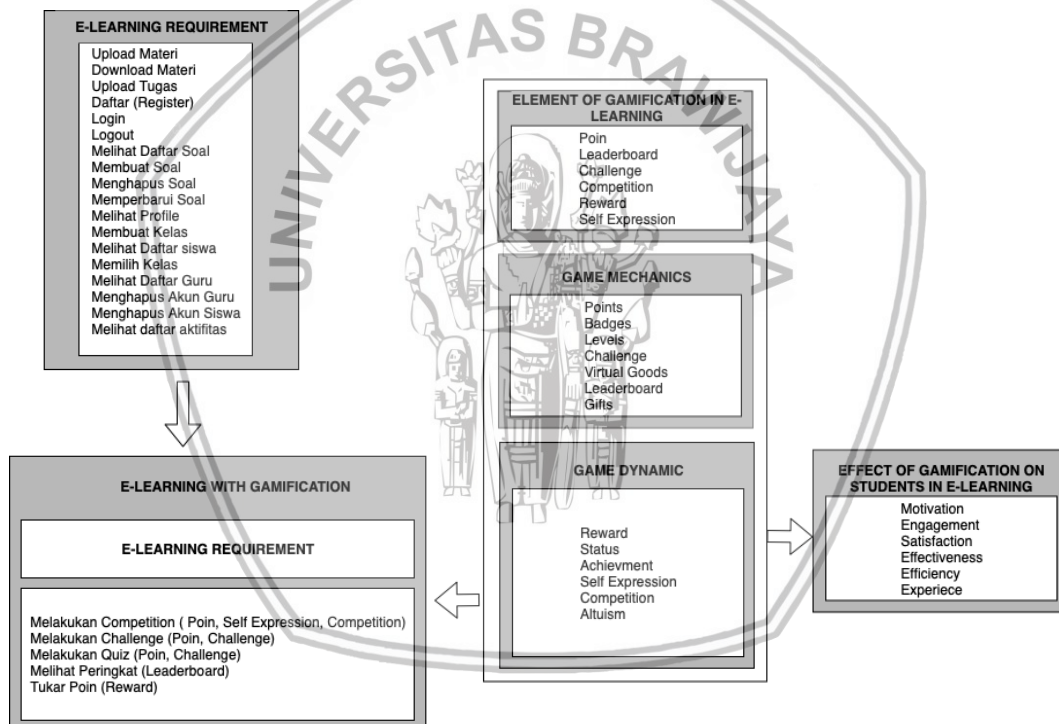
Tukar Poin	
Actor	<i>User</i>
Objective	Mengizinkan <i>User</i> untuk bisa mengakses halaman <i>contact</i> .
Pre-Condition	Halaman depan aplikasi sudah terbuka
Main Flow	<ol style="list-style-type: none"> 1. <i>User</i> memilih menu <i>contact</i>. 2. Sistem menampilkan halaman <i>contact</i>.
Alternative Flow	-
Post -Condition	<i>User</i> berhasil mengakses halaman <i>contact</i> .

BAB 5 PERANCANGAN SISTEM

Pada bagian ini merupakan penjelasan bagaimana proses perancangan yang dilakukan dan akan menghasilkan beberapa diagram dan perancangan lainnya yaitu, arsitektur penggabungan *gamification* dan *e-learning*, pemodelan *sequence diagram*, *class diagram*, perancangan basis data, perancangan komponen dan juga perancangan antarmuka. Pada perancangan sistem akan dijelaskan secara detail dengan tujuan agar dapat mudah diimplementasikan kedalam bahas program.

5.1 Arsitektur *Gamification* dalam *E-learning*

Arsitektur yang akan dijelaskan pada sub bab ini adalah komponen-komponen bagian dari *e-learning* dan juga *gamification*. Arsitektur *e-learning* dengan *gamification* dijelaskan dalam Gambar 5.1.



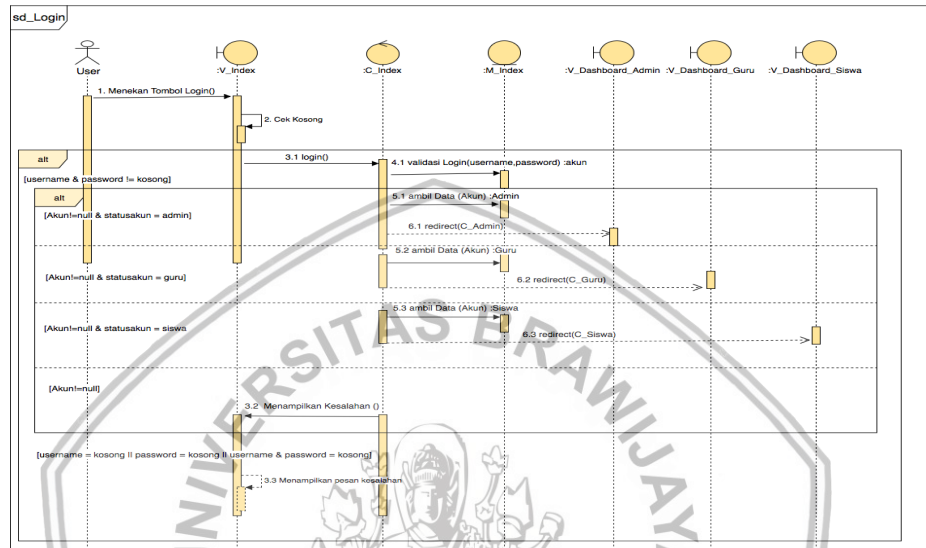
Gambar 5.1 Arsitektur *E-learning* Dengan *Gamification*

5.2 Pemodelan *Sequence diagram*

Sequence diagram adalah sebuah langkah-langkah atau tahapan proses untuk mencapai sebuah tujuan. Didalam *sequence diagram* terdapat beberapa bagian penting yaitu interaksi, objek, proses, tiga bagian tersebut akan memiliki hubungan satu dengan yang lainnya sehingga dapat menjalankan tahapan yang seharusnya untuk mencapai tujuan yang diinginkan. lima sample *sequence diagram* yang akan dijelaskan yaitu *sequence diagram login*, membuat soal, membuat kelas, menghapus akun guru, melihat *profile* siswa.

5.2.1 Login

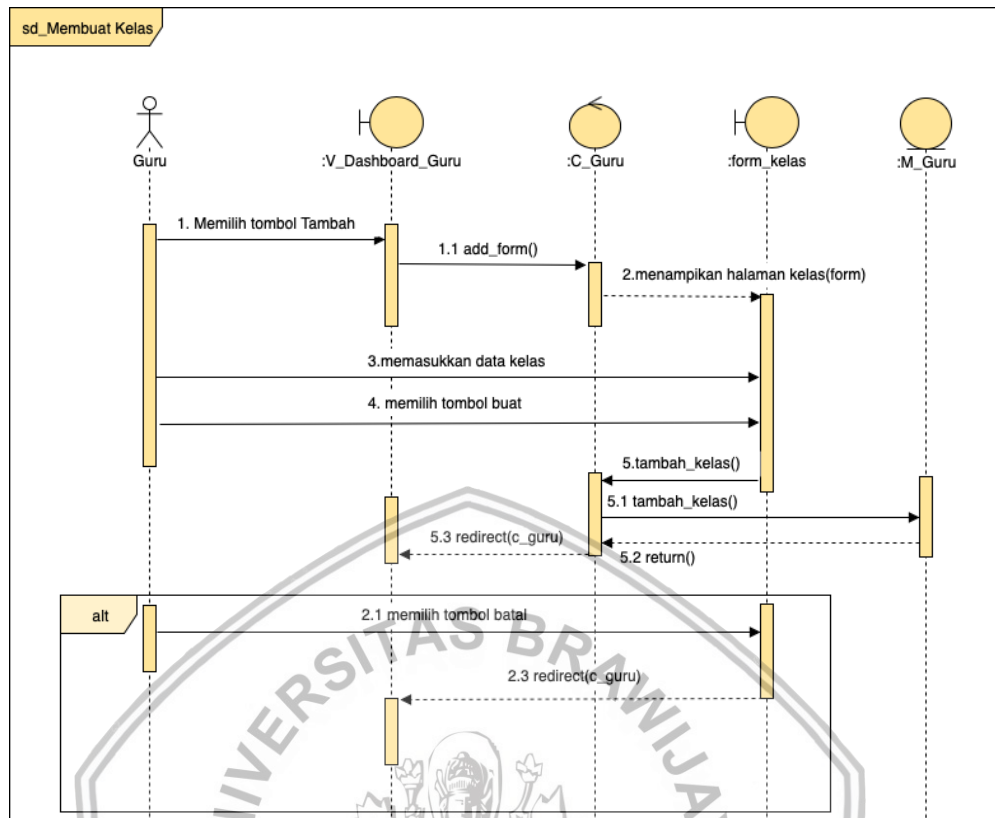
Untuk mencapai tujuan berhasil melakukan *login* tahapan dijelaskan dalam Gambar 5.1 dimana *user* sebagai aktor belum teridentifikasi akan menjalankan operasi *login()* kemudian akan dicek data masukan *user* yang nantinya akan dijadikan acuan oleh sistem untuk menentukan *user* tersebut memiliki *role* sebagai admin, guru, ataupun siswa. Setelah itu sistem akan menampilkan halaman depan sesuai dengan *role* yang dideteksi oleh sistem.



Gambar 5.2 Sequence diagram login

5.2.2 Membuat Kelas

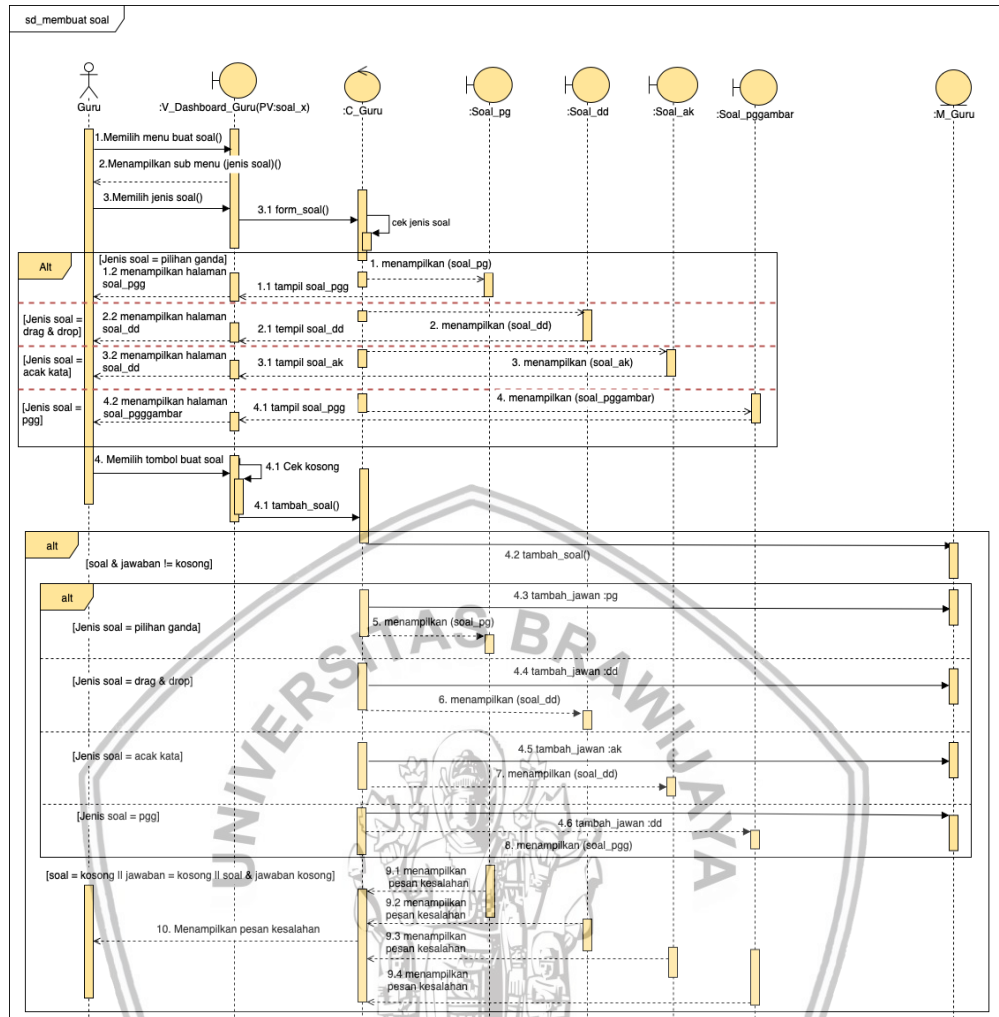
Dalam Gambar 5.2 merupakan penjelasan tahapan untuk mencapai tujuan yaitu berhasil membuat kelas, sistem akan secara otomatis menjalankan proses *add_form* ketika guru memilih tombol tambah. Dengan dijalankan proses *add_form* sistem akan secara otomatis menampilkan halaman tambah kelas, langkah selanjutnya adalah guru memberikan masuka data kelas kedalam sistem dan sistem akan menjalankan operasi *tambah_kelas*, jika proses tambah kelas berhasil sistem akan secara otomatis menampilkan halaman depan guru. Dalam Gambar 5.2 tersebut juga terdapat kemungkinan lain yaitu ketika guru memilih tombol batal pada halaman tambah kelas, sistem akan secara otomatis menampilkan halaman depan guru.



Gambar 5.3 Sequence diagram membuat kelas

5.2.3 membuat soal

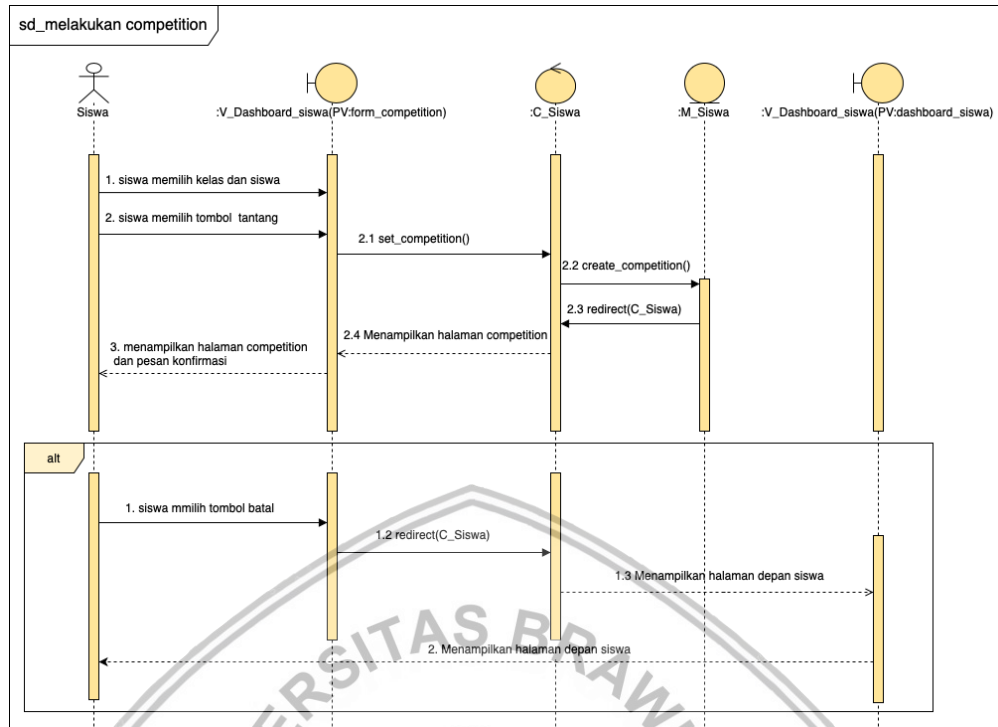
Dalam Gambar 5.3 merupakan *sequence diagram* yang menjelaskan tahapan dimana guru berhasil membuat soal. Langkah-langkah yang harus dilakukan aktor guru adalah memilih menu buat soal, kemudia memilih jenis soal yang akan dibuat. Ada empat jenis soal yang disediakan sistem untuk guru yaitu pilihan ganda, acak kata, *drag & drop*, dan pilihan ganda dalam bentuk gambar. Setelah memilih jenis soal *controller* C_Guru akan memeriksa jenis soal yang diperiksa oleh guru melalui fungsi `form_soal()`, kemudian sistem akan menampilkan halaman sesuai dengan pilihan jenis soal guru. Langkah selanjutnya setelah guru memasukkan data soal dan memilih tombol buat soal. Jika *form* soal tidak ada yang kosong maka *controller* C_Siswa akan memanggil fungsi `tambah_soal()` pada model M_Guru dan akan menambahkan data soal kedalam *database* sesuai dengan jenis soalnya. jika kosong akan menampilkan pesan error.



Gambar 5.4 *sequence diagram* membuat Soal

5.2.4 Melakukan *competition*

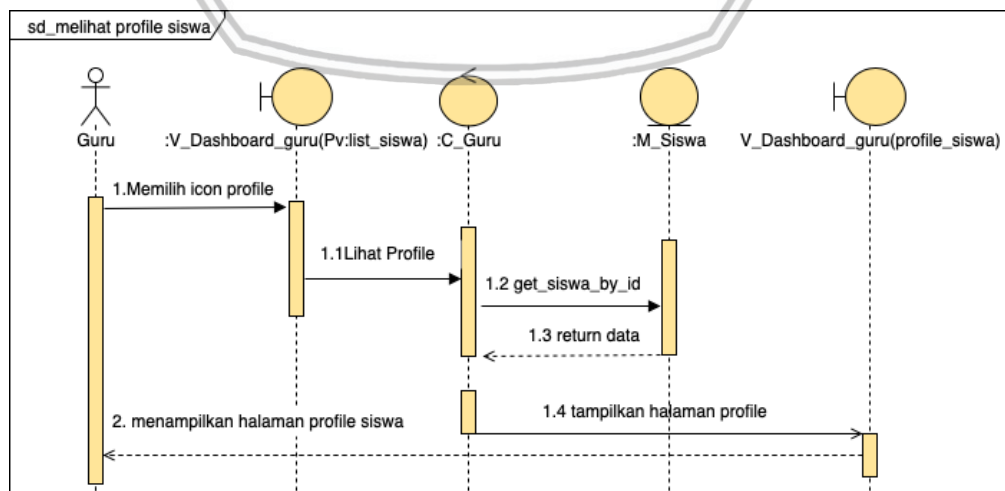
Sequence diagram dalam Gambar 5.4 menjelaskan tahapan untuk melakukan *competition*. Pertama siswa memilih kelas dan siswa yang akan ditantang, kemudian siswa memilih tombol tantang. Setelah siswa memilih tombol tantang secara otomatis *controller* C_Siswa akan menjalankan sebuah fungsi *set_competition()*, dimana didalam *controller* tersebut akan memanggil fungsi pada *model* M_Siswa yaitu *create_competition*. Setelah fungsi model tersebut dijalankan langkah selanjutnya yaitu akan mengalihkan kembali kehalaman *competition* beserta pesan konfirmasi bawah proses menantang siswa lain sudah berhasil dilakukan.



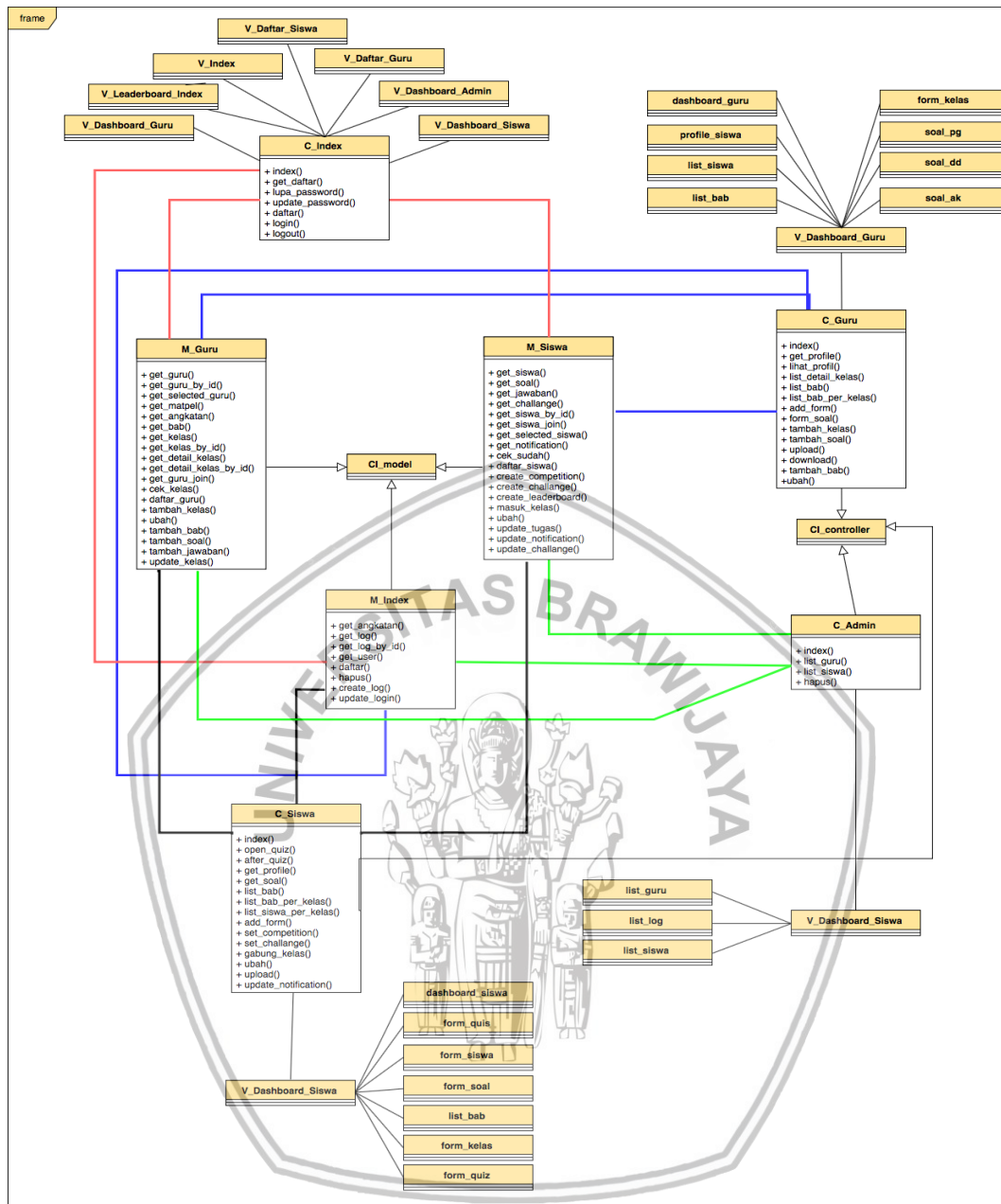
Gambar 5.5 Sequence diagram melakukan competition

5.2.5 Melihat Profile Siswa

Sequence diagram melihat profile siswa dalam Gambar 5.5 merupakan alur kontrol yang menjelaskan tahapan kejadian pada fungsi melihat *profile* siswa. Pertama guru memilih tombol *icon* pada siswa tertentu didalam tabel, *controller* C_Guru akan memanggil fungsi lihat_profile(), kemudian fungsi tersebut akan memanggil *model* M_siswa dimana terdapat fungsi untuk mengambil data siswa yaitu get_siswa_by_id, setelah data dari *model* berhasil dikirimkan ke *controller* maka secara otomatis akan menampilkan halaman profile siswa. Seperti Dalam Gambar 5.5.



Gambar 5.6 Sequence diagram melihat profile siswa

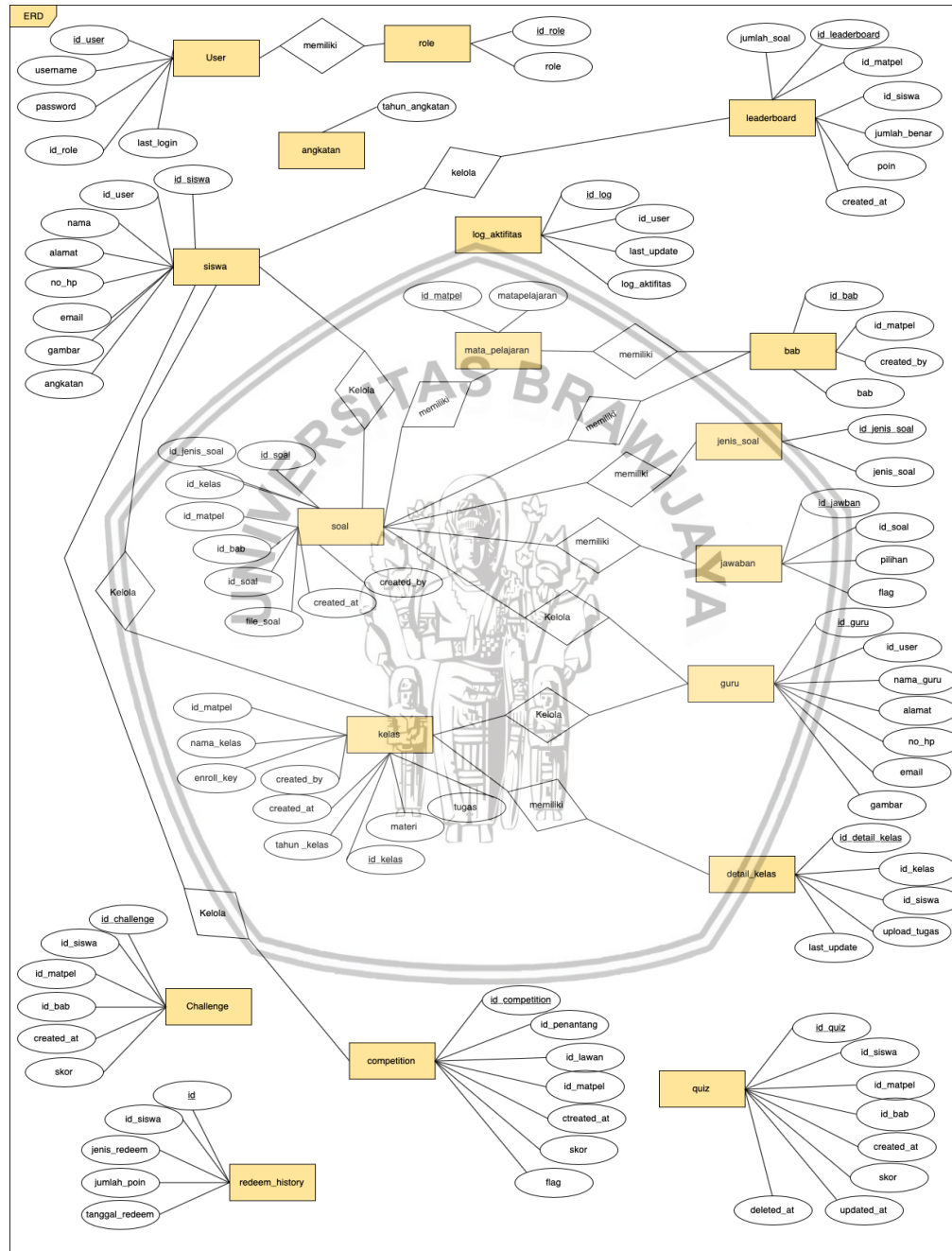


Gambar 5.7 Class diagram

5.3 Pemodelan Class Diagram

class diagram bertujuan untuk mendeskripsikan objek-objek yang terbentuk dan relasi diantara objek-objek tersebut. Terdapat dua hubungan pada *class diagram* dalam Gambar 5.6, yaitu hubungan asosiasi dan juga inheritance. Asosiasi pada gambar diatas adalah garis yang tidak putus-putus dan juga tidak memilih panah pada ujung garisnya, sedangkan inheritance pada gambar diatas adalah garis yang tidak putus dan juga memiliki panah pada ujung garisnya.

Perancangan basis data menggunakan *Entity Relationship Diagram* (ERD), dalam perancangan ini menjelaskan hubungan atau interaksi antar *entity* satu dengan *entity* lainnya. diagram ini juga dapat menjelaskan atau mengidentifikasi ciri-ciri apa saja yang terdapat pada sebuah entitas seperti dalam Gambar 5.7.



Gambar 5.8 Entity relationship diagram (ERD)

5.5 Perancangan Komponen

Perancangan komponen mendefinisikan struktur data atau algoritme yang digunakan dalam suatu komponen perangkat lunak. Didalam perancangan

komponen ini hanya mencantumkan algoritme untuk guru menambahkan kelas sehingga tidak semua algoritme dicantumkan. Algoritme tambah kelas ini akan diletakkan dalam *controller* C_Guru, serta *model* yang mendukung jalannya fungsi tersebut yaitu tambah_kelas pada *model* M_Guru dan create_log pada *model* M_index untuk memasukkan data kelas kedalam database beserta data aktifitasnya.

5.5.1 Controller C_Guru

tambah_kelas() merupakan proses yang diambil dari *controller* C_Guru, proses tersebut adalah fungsi untuk menambahkan kelas pada aplikasi *e-learning*. Pada Tabel 5.1 dijelaskan algoritme tambah_kelas()

Tabel 5.1 Algoritme fungsi tambah_kelas

1	Mulai
2	Memanggil model M_Index
3	Membuat variabel dengan nama Post, isinya berupa inputan dari user.
4	Membuat variabel dengan nama Date, isinya berupa waktu dengan format (Y-m-d H:i:s)
5	Membuat variabel dengan nama Path, isinya berupa lokasi penyimpanan file.
6	Membuat variable array dua dimensi untuk menyimpan semua informasi dari file yang akan diupload secara sementara.
7	Membuat variable array satu dimensi (\$config) untuk proses upload yang berupa tempat file disimpan, tipe file yang diterima sistem, dan juga nama file ketika disimpan.
8	Memanggil <i>library upload</i> , dengan isi variabel array satu dimensi(\$config) tadi.
9	<p>Jika do_upload memiliki isi dengan nama gambar maka akan menjalankan :</p> <ul style="list-style-type: none"> • Membuat variabel dengan nama upload_data dengan isi data upload. • Membuat variabel dengan nama gambar dengan isi perpaduan antara variabel path dengan variabel upload_data dengan isi nama file(file_name). • Membuat variabel dengan nama kelas dengan isi perpaduan antara inputan user berupa nama dan angkatan. • Membuat variabel dengan nama foto, isinya berupa inputan user berupa gambar. • Membuat variabel array dengan nama data dimana didalamnya terdapat id_matpel, nama_kelas, enroll_key, created_by, created_at, tahun_kelas, keterangan, gambar_kelas. • Membuat variabel dengan nama tambah, isinya berupa fungsi untuk menjalankan <i>model</i> M_Guru dengan fungsi tambah_kelas dengan variabel data.
10	jika variabel tambah tidak sama dengan nol maka akan menjalankan :

	<ul style="list-style-type: none"> Membuat data array dengan nama <i>data1</i>, isinya berupa <i>id_user</i>, <i>last_update</i>, <i>log_aktifitas</i>. Membuat variabel dengan nama <i>log</i>, isinya berupa fungsi untuk menjalankan <i>model M_Index</i> dengan fungsi <i>create_log</i> dengan variabel <i>data1</i>. Mengalihkan ke <i>controller C_Guru</i>. <p>Jika tidak sama dengan nol maka akan menjalankan:</p> <ul style="list-style-type: none"> Menampilkan pesan <i>error</i> Mengalihkan ke <i>controller C_Guru</i>
11	<p>Jika <i>do_upload</i> isinya bukan gambar maka akan menjalankan :</p> <ul style="list-style-type: none"> Menampilkan pesan <i>error</i> Mengalihkan ke <i>controller C_Guru</i> dan menjalankan fungsi <i>add_form</i>
12	Selesai

5.5.2 Model M_Guru

tambah_kelas() merupakan proses yang diambil dari *model M_Guru*, proses tersebut adalah fungsi untuk menambahkan data kelas kedalam *database* pada aplikasi *e-learning*. Pada Tabel 5.2 dijelaskan algoritme *tambah_kelas()*

Tabel 5.2 Algoritme fungsi *tambah_kelas*

1	Mulai
2	Membuat variabel dengan nama <i>hasil</i> , isinya berupa fungsi untuk memasukkan data kedalam <i>database</i> yaitu <i>insert</i> dengan data variabel kelas dan juga <i>value</i> .
3	Mengembalikan data variabel <i>hasil</i>
4	selesai

5.5.3 Model M_Index

Create_log() merupakan proses yang diambil dari *model M_Guru*, proses tersebut adalah fungsi untuk menambahkan data aktifitas kedalam *database* pada aplikasi *e-learning*. Pada Tabel 5.3 dijelaskan algoritme *create_log()*

Tabel 5.3 Algoritme fungsi *create_log*

1	Mulai
2	Membuat variabel array dengan nama <i>data</i> , isinya berupa <i>id_user</i> , <i>last_update</i> , <i>log_aktivitas</i> yang didalamnya adalah variabel <i>value</i>
3	Membuat variabel dengan nama <i>hasil</i> , isinya berupa fungsi untuk memasukkan data kedalam <i>database</i> yaitu <i>insert</i> dengan data <i>log</i> dan juga variabel array dengan nama <i>data</i> tadi.

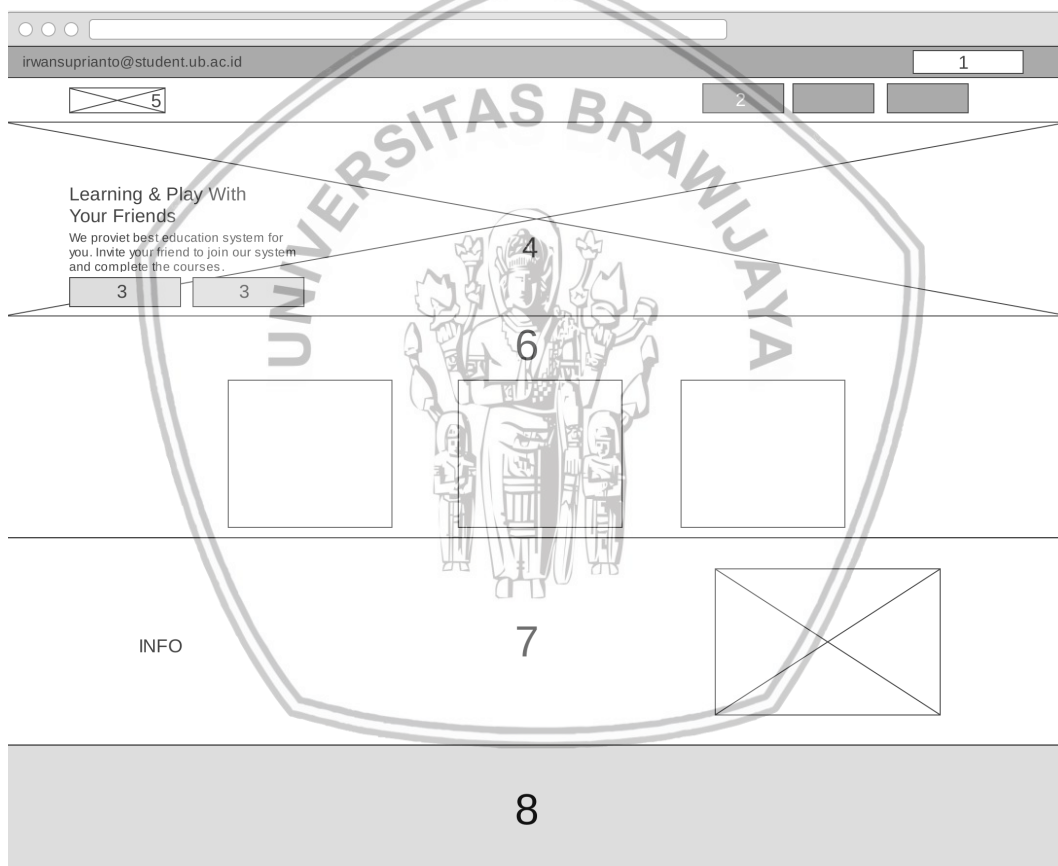
4	Mengembalikan data variabel hasil
5	Selesai

5.6 Perancangan Antarmuka

Berikut ini akan dijelaskan lima sample dari halaman web yang dibuat, yaitu halaman Depan Sistem, halaman daftar kelas, halaman daftar siswa, halaman *profile* siswa, dan halaman tukar poin.

5.6.1 Halaman Depan

Perancangan halaman depan merupakan halaman pertama *user* sebelum masuk kedalam sistem aplikasi *elearning*.



Gambar 5.9 Halaman depan

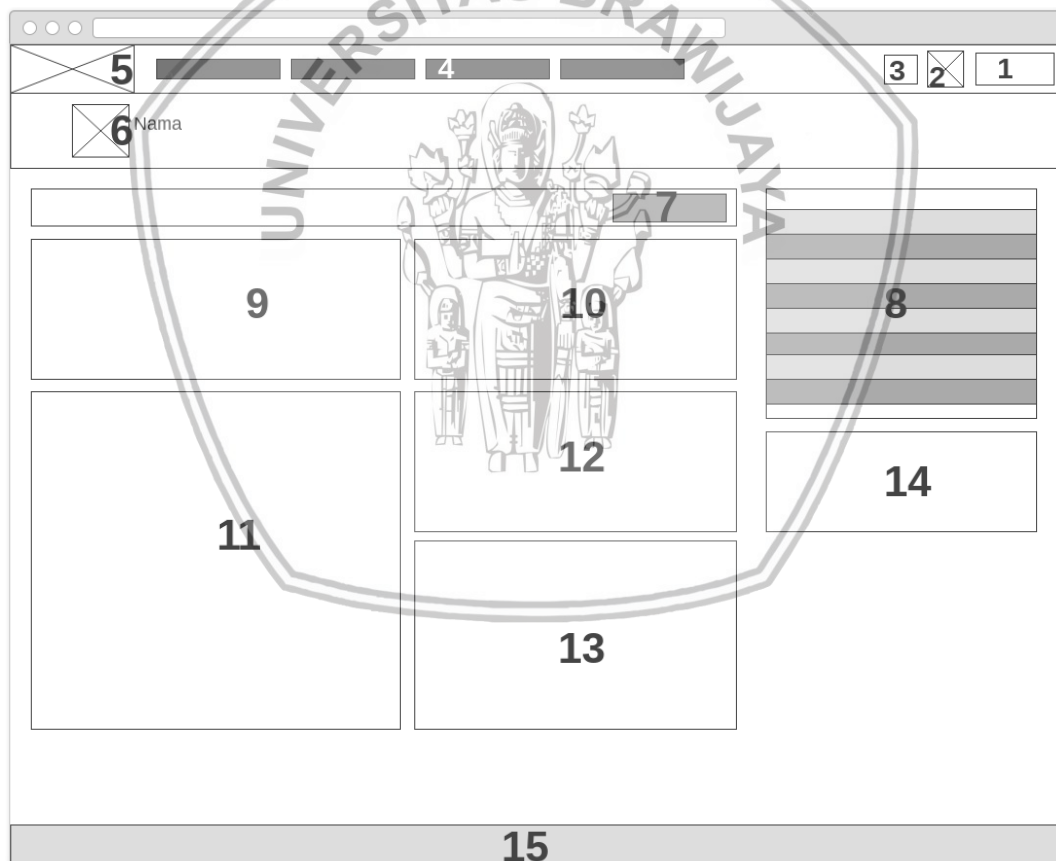
Halaman depan merupakan halaman yang pertama kali dilihat ketika mengakses aplikasi *E-learning* sebelum melakukan *login*. Berikut penjelasan detail mengenai halaman depan aplikasi *E-learning*.

1. Nomor satu pada gambar diatas adalah sebuah tombol *login*, yang terdapat pada bagian kanan atas header. Login termasuk dalam kelomkok header.
2. Nomor dua adalah menu navigasi yang termasuk dalam kelompok header, terdapat tiga menu yaitu *home*, *about*, *contact*.

3. Sedangkan nomor tiga adalah tombol untuk daftar, terdapat dua tombol daftar pada gambar diatas, daftar sebagai siswa dan daftar sebagai guru.
4. Sedangkan nomor empat adalah gambar yang digunakan sebagai latarbelakang, gambar ini termasuk dalam kelompok header.
5. Logo dan nama aplikasi, nomor lima yang ada pada gambar diatas adalah sebuah grup yang didalamnya terdapat logo dan nama aplikasi.
6. Nomor enam termasuk dalam kelompok isi halaman depan yaitu peringkat yang menampilkan informasi tiga peringkat teratas siswa.
7. Nomor tujuh juga termasuk kedalam kelompok isi halaman depan, berupa info-info dari aplikasi *E-learning* ini.
8. Sedangkan nomor delapan adalah *footer*, berisi informasi-informasi dan juga *quick link* yang mempermudah user berpindah halaman.

5.6.2 Halaman Depan siswa

Perancangan halaman depan siswa adalah halaman pertama yang akan dilihat siswa ketika berhasil masuk kedalam sistem.



Gambar 5.10 Halaman depan siswa

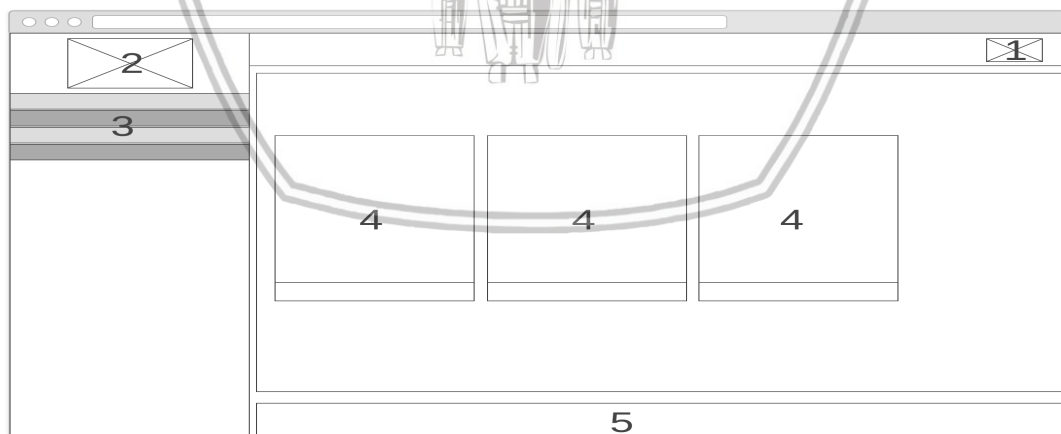
Dibawah ini adalah penjelasan terkait dengan gambar yang ada diatas.

1. Tombol keluar, tombol ini digunakan siswa ketika ingin keluar dari sistem.
2. Nomor dua adalah gambar profile siswa yang dimana didalamnya terdapat beberapa menu

3. Notifikasi, adalah bagian yang memberikan informasi terkait dengan informasi request *competition* dari siswa lain.
4. Menu navigasi satu, adalah menu yang digunakan siswa untuk berpindah-pindah halaman.
5. Logo dan nama aplikasi
6. Profil siswa, berisi gambar siswa, nama siswa dan juga link untuk berpindah halaman ke *profile* siswa.
7. Tombol tambah kelas, tombol ini digunakan siswa untuk menambahkan atau masuk kedalam kelas yang sudah dibuat oleh guru sebelumnya.
8. Menu navigasi dua, adalah menu yang isinya lebih lengkap dari menu navigasi satu. Sedangkan fungsinya sama dengan menu navigasi satu.
9. Daftar kelas, pada kolom ini berisi daftar beberapa kelas yang sudah siswa tambahkan.
10. Penghargaan, pada kolom ini berisi penghargaan yang sudah dimiliki siswa
11. Peringkat, pada kolom ini menampilkan tiga peringkat teratas siswa berdasarkan poin.
12. Statistik, kolom ini menjelaskan jumlah poin yang dimiliki siswa baik itu keseluruhan atau untuk setiap mata jawaban.
13. Riwayat penukaran, adalah kolom yang berisi daftar-daftar penukaran poin yang dilakukan siswa.
14. *Slider* kelas, adalah kelas yang ditampilkan dalam bentuk menu geser.
15. *Footer*, adalah bagian bawah dari tampilan halaman depan siswa.

5.6.3 Halaman Depan Guru

Perancangan halaman depan siswa adalah halaman pertama yang akan dilihat siswa ketika berhasil masuk kedalam sistem.



Gambar 5.11 Halaman depan guru

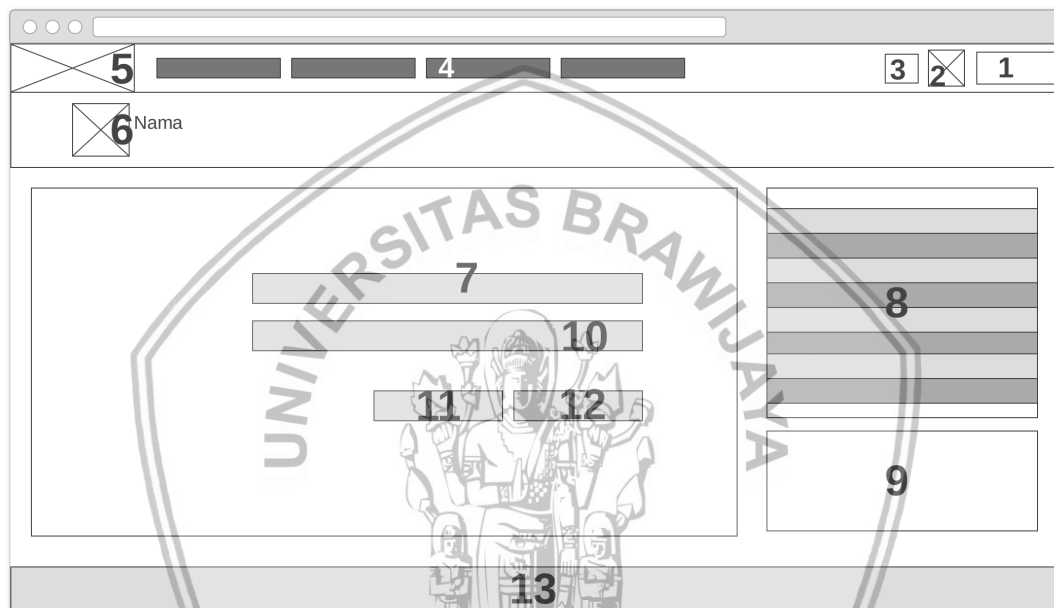
Gambar diatas merupakan bagian-bagian dari halaman depan guru yang akan dijelaskan dibawah ini sesuai dengan nomor yang diberikapan pada gambar.

1. Nomor satu adalah gambar profile guru dan terdapat beberapa menu didalamnya.
2. Nomor dua adalah gambar atau logo beserta nama aplikasi *E-learning*.

3. Nomor tiga adalah menu navigasi yang akan digunakan siswa untuk berpindah-pindah halaman.
4. Nomor empat adalah isi dari halaman depan guru yang berupa daftar kelas yang sudah dibuat oleh guru.
5. Nomor lima adalah *footer* dari halaman depan guru.

5.6.4 Halaman *Competition*

Perancangan halaman *competition* adalah halaman yang dapat diakses oleh siswa. Pejelasan yang berkaitan dengan bagian-bagian yang terdapat pada halaman *competition* berada diwabah ini.



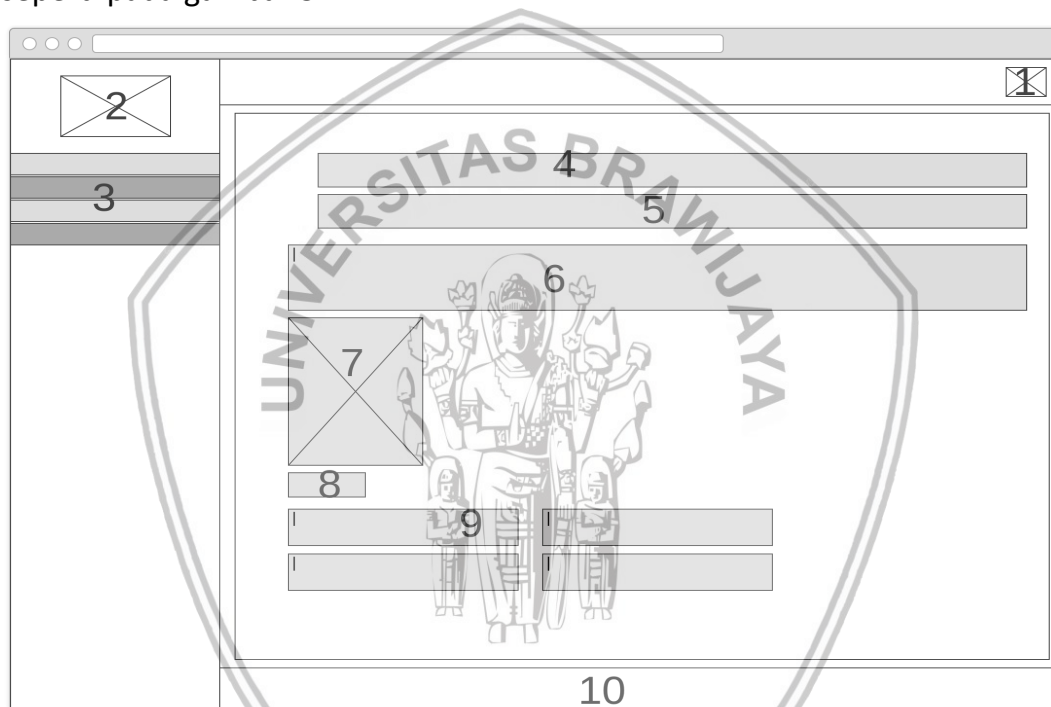
Gambar 5.12 Halaman *competition*

1. Tombol keluar, tombol ini digunakan siswa ketika ingin keluar dari sistem.
2. Nomor dua adalah gambar profile siswa yang dimana didalamnya terdapat beberapa menu
3. Notifikasi, adalah bagian yang memberikan informasi terkait dengan informasi request *competition* dari siswa lain.
4. Menu navigasi satu, adalah menu yang digunakan siswa untuk berpindah-pindah halaman.
5. Logo dan nama aplikasi
6. Profil siswa, berisi gambar siswa, nama siswa dan juga link untuk berpindah halaman ke *profile* siswa.
7. Daftar kelas, menu ini adalah menu yang digunakan siswa untuk memilih kelas siswa yang ingin ditantang.
8. Menu navigasi dua, adalah menu yang isinya lebih lengkap dari menu navigasi satu. Sedangkan fungsinya sama dengan menu navigasi satu.
9. Daftar kelas, pada kolom ini berisi daftar beberapa kelas yang sudah siswa tambahkan.

10. Daftar siswa, menu ini adalah menu yang digunakan siswa untuk memilih siswa yang ingin ditantang.
11. Tombol batal, Tombol batal adalah tombol yang digunakan siswa jika tidak jadi melakukan *competition*
12. Tombol tantang, tombol ini digunakan oleh siswa jika ingin mengirim *request* tantangan kesiswa lain.
13. *Footer*, adalah bagian bawah dari tampilan halaman depan siswa.

5.6.5 Halaman Soal *Drag & Drop*

Perancangan soal *drag & drop* adalah halaman yang digunakan guru untuk membuat soal dalam bentuk *drag & drop*. Perancangan halaman seperti pada gambar 5.12.



Gambar 5.13 Halaman soal *drag & drop*

Penjelsan yang berkaitan dengan bagian-bagian pada gambar diatas seperti berikut.

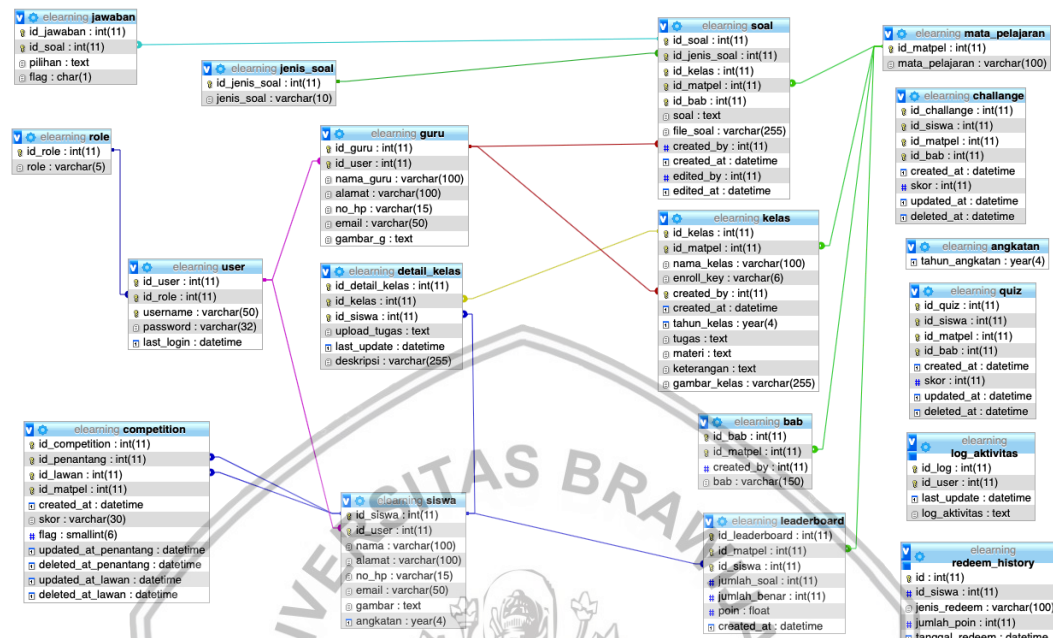
1. Nomor satu adalah gambar profile guru dan terdapat beberapa menu didalamnya.
2. Nomor dua adalah gambar atau logo beserta nama aplikasi *E-learning*.
3. Nomor tiga adalah menu navigasi yang akan digunakan siswa untuk berpindah-pindah halaman.
4. Daftar kelas, menu ini adalah menu yang digunakan guru untuk memilih kelas.
5. Daftar bab, menu ini adalah menu yang digunakan guru untuk memilih bab.
6. Kotak soal, adalah bagian yang digunakan guru untuk memasukkan soal.
7. Gamar soal, adalah tampilan gambar yang diupload oleh guru.

8. Tombol upload, tombol ini digunakan oleh guru untuk upload gambar soal *drag & drop*.
9. Kotak jawaban, kotak atau *field* jawaban adalah kotak yang digunakan untuk memasukkan jawaban soal.
10. Nomor lima adalah *footer* dari halaman depan guru.



BAB 6 IMPLEMENTASI

6.1 Implementasi Database



Gambar 6.1 Implementasi database

Implementasi database adalah lanjutan dari hasil perancangan *entity relationship diagram* (ERD) dimana hubungan dari setia tabel yang ada diatas merupakan gamabar tabel yang ada didalam *database*. tabel-tabel tersebut nantinya akan diginukan sebagai tempat menyimpan dan mengambil data untuk diproses pada aplikasi *e-learning*, dalam Gambar 6.1 menjelaskan tabel-tabel apa saja yang tesedia didalam *database* apalikasi *e-learning* beserta dengan atributnya.

6.2 Impelementasi Algoritme

Algoritme yang ditampilkan adalah algoritme tambah kelas, dengan demikian tidak semua algoritme ditampilkan dalam implementasi algoritme ini. Fungsi-fungsi yang mendukung proses tambah kelas seperti *model* pendukungnya juga akan ditampilkan dalam sub bab implementasi algoritme ini yaitu fungsi tambah_kelas() yang merupakan bagian dari *model* M_Guru dan fungsi create_log() bagian dari *model* M_Index. Proses penambahan data kelas dijalankan oleh *controller* C_Guru dengan fungsi tambah_kelas().

6.2.1 Controller C_Guru

Tambah_kelas() merupakan fungsi atau proses utuk menambahkan kelas yang di ambil dari *controller* C_Guru. Pada proses ini terdapat beberapa data atau masukan yang nantinya akan digunakan sebagai *trigger* untuk mengambil atau memasukkan data, seperti nama kelas, kode kelas(enroll

code), pembuat kelas, waktu pembuatan kelas, tahun kelas, keterangan, dan juga gambar kelas. Kemudian data-data inputan tersebut akan dimasukkan kedalam sebuah variabel yang nantinya akan dikirimkan datanya melalui fungsi model untuk menambahkan data kedalam basis data, bersamaan dengan itu akan dibuat daftar aktifitas guru melalui fungsi tambah_kelas. Algoritme tambah_kelas dapat dilihat pada Tabel 6.1.

Tabel 6.1 Algoritme fungsi tambah_kelas

1	function tambah_kelas(){
2	\$this->load->model('M_Index');
3	\$post = \$this->input->post();
4	\$date_now = date("Y-m-d H:i:s");
5	\$path .= './fl/class/';
6	\$TEMPORARY DATA;
7	\$UPLOAD_SETTINGS;
8	\$this->load->library('upload', \$config);
9	if (\$this->upload->do_upload('gambar')) {
10	\$upload_data = \$this->upload->data();
12	\$gambar = \$path . \$upload_data['file_name'];
13	\$kelas = \$post["nama"] . ' - ' .
14	\$post["angkatan"];
15	\$foto = \$this->input->post('gambar');
16	\$data = array('id_matpel'=> \$post['matpel'],
17	'nama_kelas'=>\$kelas,
18	'enroll_key'=> \$post['kode'],
19	'created_by'=> \$post['id_guru'],
20	'created_at'=> \$date_now,
21	'tahun_kelas'=> \$post['angkatan'],
22	'keterangan'=> \$post['keterangan'],
23	'gambar_kelas'=> \$gambar);
24	\$tambah = \$this->M_Guru->tambah_kelas(\$data);
25	if (\$tambah != 0) {
26	\$datal = array('id_user'=> \$this->session
27	>userdata('id_user'),
28	'last_update'=> \$date_now,
28	'log_aktivitas'=> \$this->session-
	>userdata('nama') . ' Membuat
	Kelas '. \$kelas);
29	\$log = \$this->M_Index->create_log(\$datal);
30	redirect('C_Guru');
	}else{
31	\$this->session->set_flashdata('flash_message',
32	warn_msg('Gagal Membuat Kelas Baru'));
33	redirect('C_Guru');
	}
	}else{
34	\$this->session->set_flashdata('flash_message',
	warn_msg(\$this->upload->display_errors()));
35	redirect('C_Guru/add_form/');
36	}
37	}

6.2.2 Model M_Guru

Fungsi tambah_kelas() pada *model* M_Guru berisi *query* untuk menambahkan data kelas kedalam basis data, fungsi ini ditunjukkan pada Tabel 6.2.

Tabel 6.2 Algoritme fungsi tambah_kelas

1	function tambah_kelas(\$value){
2	\$hasil = \$this->db->insert(\$this->kelas,\$value);
3	return \$hasil;
4	}

6.2.3 Model M_Index

Fungsi create_log() pada *model* M_Index berisi *query* untuk menambahkan data aktifitas guru kedalam basis data, data yang ditambahkan berupa id_user, last_update, log_aktifitas. fungsi ini ditunjukkan pada Tabel 6.3.

Tabel 6.3 Algoritme Fungsi create_log

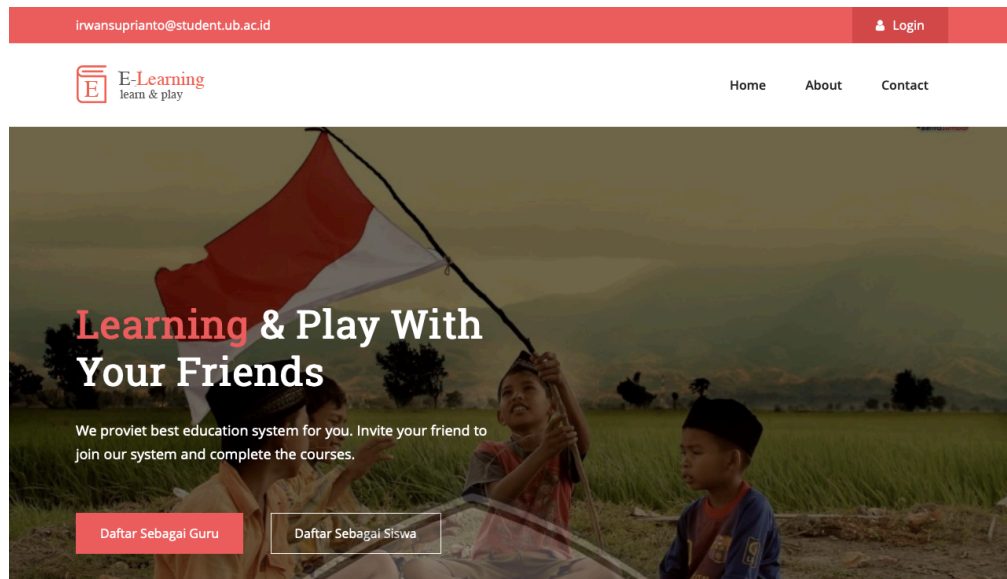
1	function create_log(\$value){
2	\$data = array('id_user' => \$value['id_user'],
3	'last_update'=> \$value['last_update'],
4	'log_aktifitas'=>\$value['log_aktifitas']);
5	\$hasil = \$this->db->insert(\$this->log, \$data);
6	return \$hasil;
7	}

6.3 Impelementasi Antarmuka

Berikut ini merupakan hasil implementasi antarmuka dari aplikasi *E-learning*. Implementasi antarmuka yang akan ditampilkan adalah halaman depan aplikasi *E-learning*, halaman login, halaman depan siswa, halaman depan guru, halaman soal pilihan ganda(pg), halaman kelas, halaman aktifitas siswa, halaman soal *challenge*, halaman peringkat, halaman *competition*. Sehingga tidak semua implementasi antarmuka ditampilkan.

6.3.1 Halaman depan Aplikasi

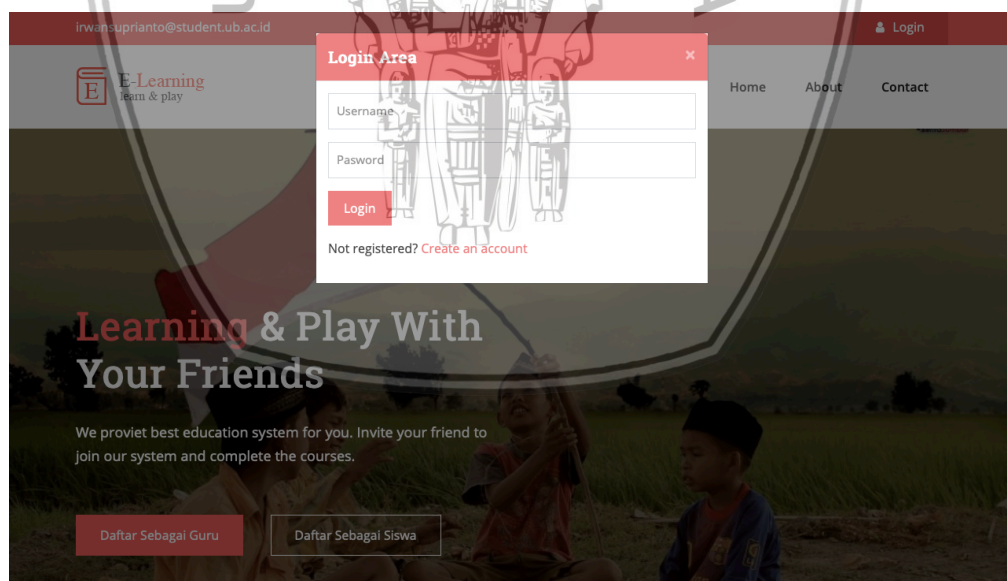
Halaman depan aplikasi merupakan halaman yang akan pertama kali pengguna lihat ketika mengakses aplikasi *e-learning* sebelum melakukan login. Pada halaman ini terdapat beberapa informasi seperti peringkat tiga teratas siswa, informasi alasan memilih aplikasi, dan terdapat informasi *footer* berisi *quick* link dan kontak. Dalam Gambar 6.2 menampilkan halaman aplikasi.



Gambar 6.2 Halaman depan aplikasi

6.3.2 Halaman Login

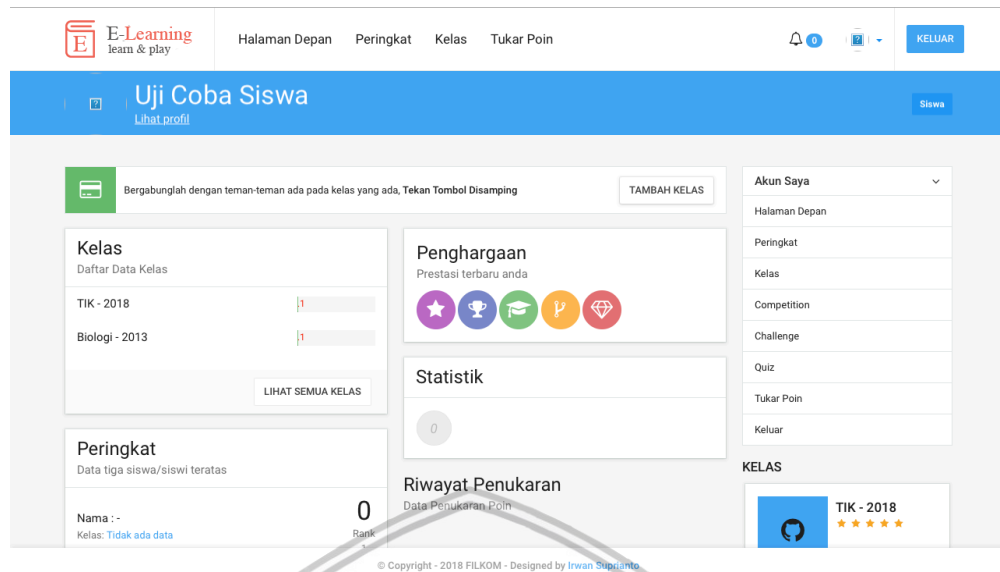
Halaman *login* merupakan halaman yang pengguna akses ketika ingin masuk kedalam sistem, ada dua kolom yang harus diisi sebelum masuk kedalam sistem yaitu *username* dan *password*. Dalam Gambar 6.3 menampilkan halaman *login*.



Gambar 6.3 Halaman login

6.3.3 Halaman Depan Siswa

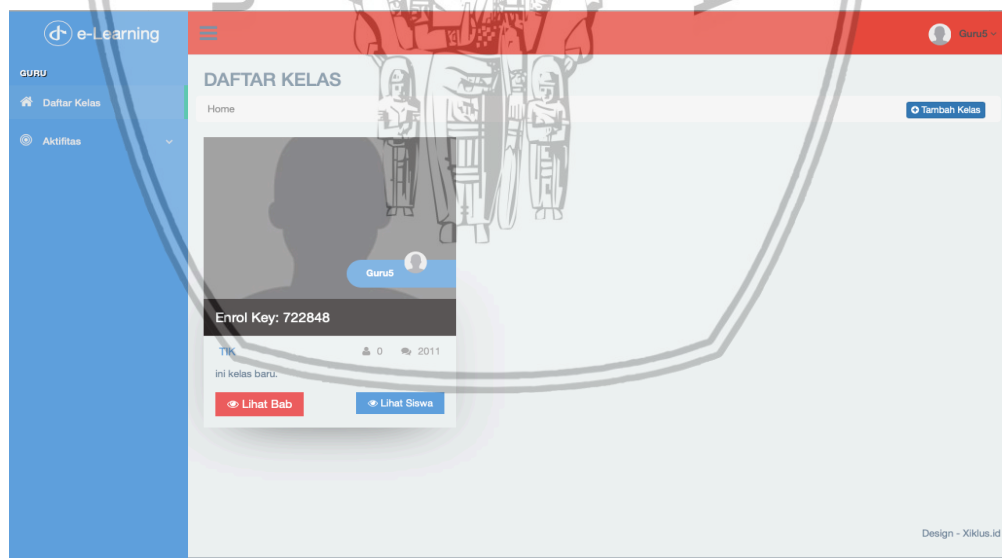
Halaman depan siswa merupakan halaman yang akan tampil ketika berhasil *login* sebagai siswa. Halaman ini menampilkan beberapa informasi seperti daftar kelas, penghargaan, statistik, peringkat, riwayat penukaran, dan juga menu navigasi halaman. Dalam Gambar 6.4 menampilkan halaman depan siswa.



Gambar 6.4 Halaman depan siswa

6.3.4 Halaman Depan Guru

Halaman depan guru merupakan halaman yang akan tampil ketika berhasil *login* sebagai guru. Halaman ini menampilkan beberapa informasi seperti daftar kelas, dan juga menu navigasi halaman. Dalam Gambar 6.5 menampilkan halaman depan guru.

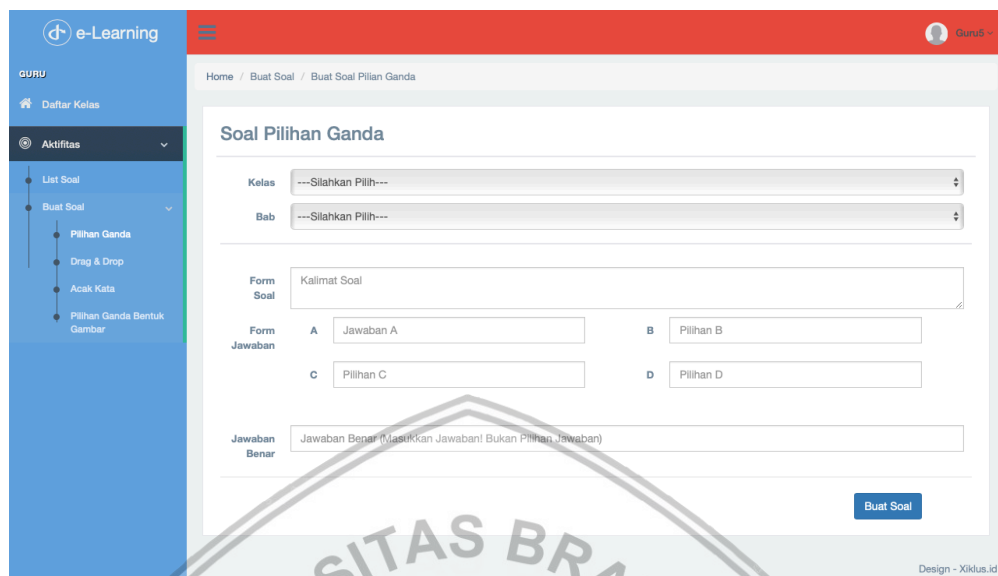


Gambar 6.5 Halaman depan guru

6.3.5 Halaman Buat Soal

Halaman buat soal merupakan halaman yang akan tampil ketika berhasil *login* sebagai guru dan mengakses halaman buat soal. Halaman ini menampilkan beberapa informasi seperti form soal yang berisi menu *dropdown* kelas, menu *dropdown* bab, kolom soal, kolom pilihan jawaban,

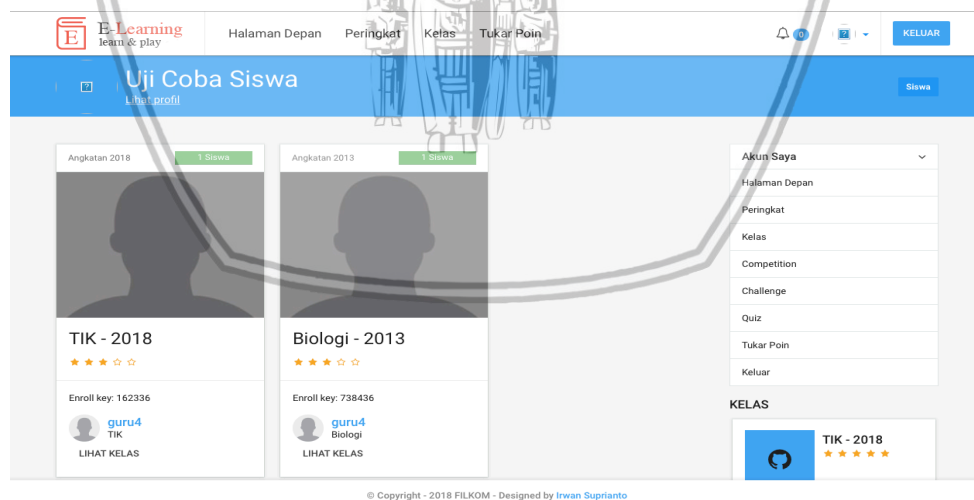
kolom jawaban benar dan juga menu navigasi halaman. Dalam Gambar 6.6 menampilkan halaman buat soal.



Gambar 6.6 Halaman buat soal

6.3.6 Halaman Kelas

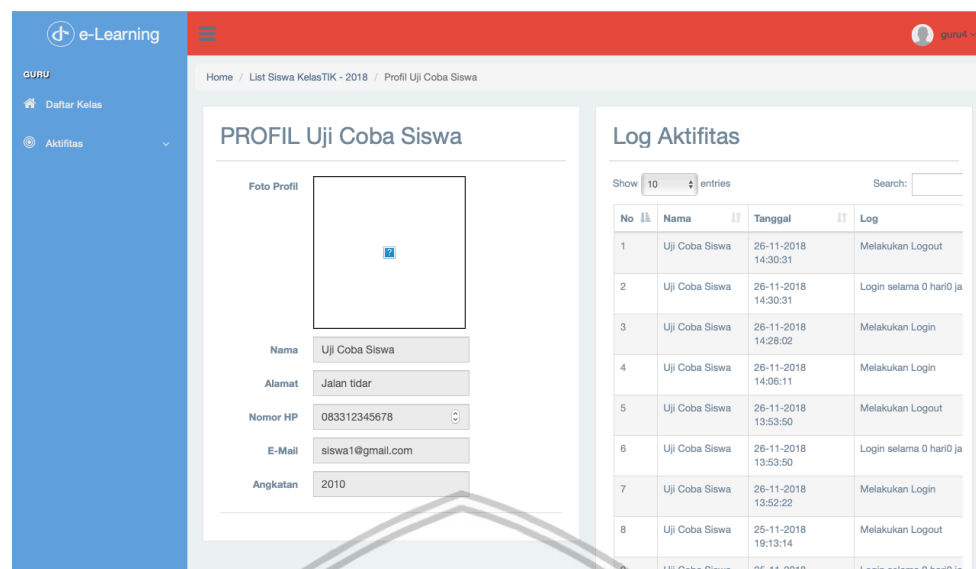
Halaman kelas merupakan halaman yang akan tampil ketika berhasil *login* sebagai siswa dan mengakses halaman kelas. Halaman ini menampilkan beberapa informasi seperti daftar kelas dan juga menu navigasi halaman. Dalam Gambar 6.7 menampilkan halaman kelas.



Gambar 6.7 Halaman kelas

6.3.7 Halaman Aktifitas Siswa

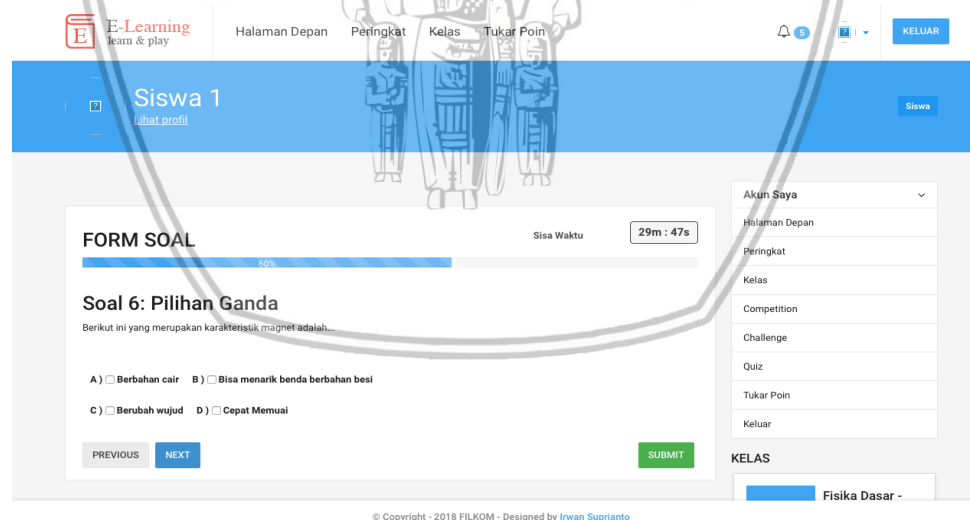
Halaman aktifitas siswa adalah halaman yang akan tampil ketika guru mengakses halaman profil siswa. Halaman ini menampilkan beberapa informasi seperti profil, daftar aktifitas siswa dan juga menu navigasi halaman. Dalam Gambar 6.8 menampilkan halaman aktifitas siswa.



Gambar 6.8 Halaman aktifitas siswa

6.3.8 Halaman Soal

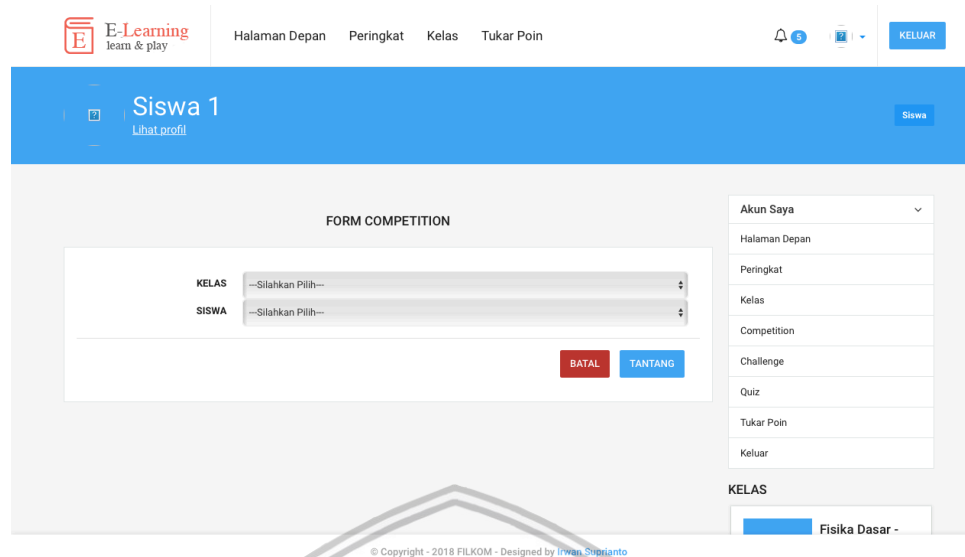
Halaman soal yaitu halaman yang akan tampil ketika siswa mengakses *competition*, *challenge*, dan *quiz*. Halaman ini menampilkan beberapa informasi seperti *progress bar*, waktu, soal, pilihan soal, tombol *submit*, *next*, *previous* dan juga menu navigasi. Dalam Gambar 6.9 menampilkan halaman soal.



Gambar 6.9 Halaman soal

6.3.9 Halaman Competition

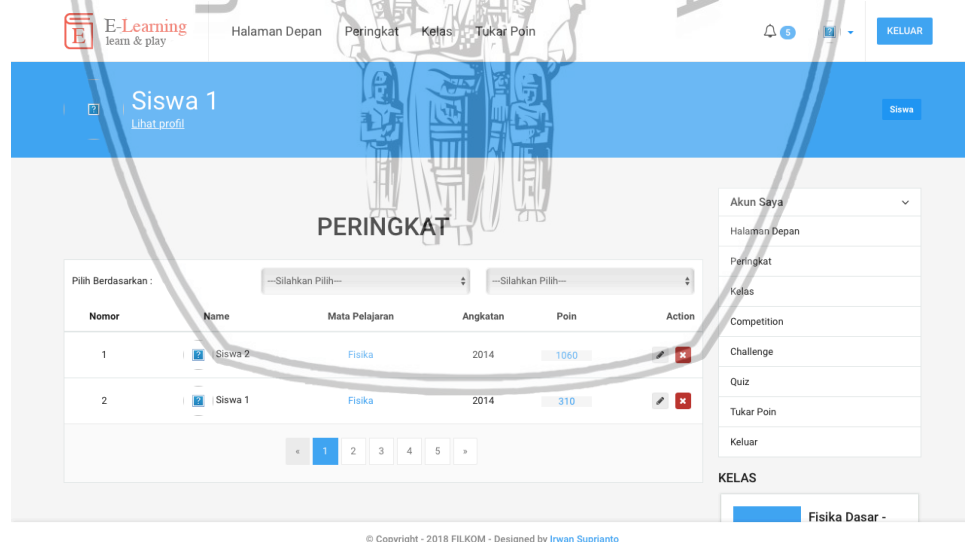
Halaman *Competition* adalah halaman yang akan tampil ketika siswa mengakses *competition*. Halaman ini menampilkan beberapa informasi seperti *dropdown* menu kelas, *dropdown* menu siswa, menu navigasi halaman, tombol batal dan juga tantangan. Dalam Gambar 6.10 menampilkan halaman *competition*.



Gambar 6.10 Halaman *competition*

6.3.10 Halaman Peringkat

Halaman peringkat adalah halaman yang akan tampil ketika siswa mengakses menu peringkat. Halaman ini menampilkan beberapa informasi seperti tabel yang berisi data siswa. Dalam Gambar 6.11 menampilkan halaman peringkat.



Gambar 6.11 Halaman peringkat

BAB 7 PENGUJIAN DAN ANALISIS

Tahap pengujian dilakukan setelah implementasi selesai dilakukan. Pengujian dilakukan untuk memeriksa apakah hasil implementasi sesuai dengan analisis kebutuhan dan perancangan atau tidak. Pengujian yang akan dilakukan adalah pengujian usnit, pengujian integrasi, dan pengujian validasi.

7.1 Pengujian *Whitebox*

Pengujian unit dilakukan menggunakan *basis path testing* yang merupakan salah satu jenis pengujian dari *white box testing*. Pengujian unit ini dilakukan untuk menguji klas-klas yang sudah didefinisikan sebelumnya. Scenario pengujian dilakukan dengan pengambilan tiga sampel uji yaitu klas C_Guru, C_Index, dan C_Siswa.

7.1.1 Pengujian Unit Klas C_Index

Operasi login()

Tabel 7.1 Algoritme login

1	public function login(){	
2	\$last_login = date("Y-m-d H:i:s");	
3	\$u = \$this->input->post('username');	1
4	\$p = \$this->input->post('password');	
5	\$this->load->model('M_Index');	
6	\$this->load->model('M_Guru');	
7	\$this->load->model('M_Siswa');	
8	\$this->load->library('form_validation');	
9	\$this->form_validation->set_rules('username', 'Username',	
10	trim required');	
11	\$this->form_validation->set_rules('password', 'Password',	
12	required trim');	
13	if (\$this->form_validation->run() == FALSE) {	3
14	\$this->session->set_flashdata('flash_message',	
15	warn_msg(validation_errors()));	4
16	redirect('C_Index');	
17	}	5
18	else{	6
19	\$hasil = \$this->M_Index->get_user(\$u,md5(\$p));	7
20	foreach (\$hasil as \$key => \$value) {	
21	\$id_user = \$value->id_user;	
22	\$role = \$value->id_role;	
23	\$username = \$value->username;	9
24	}	
25	if (!empty(\$hasil)) {	10
26	switch (\$role) {	11
27	case 1:	12
28	\$update = \$this->M_Index->update_login(\$last_login, \$id_user);	
29	\$data = array('id_user'=> \$id_user,	
30	'last_update'=> \$last_login,	
31	'log_aktivitas'=> \$username . ' Melakukan	
32	login');	

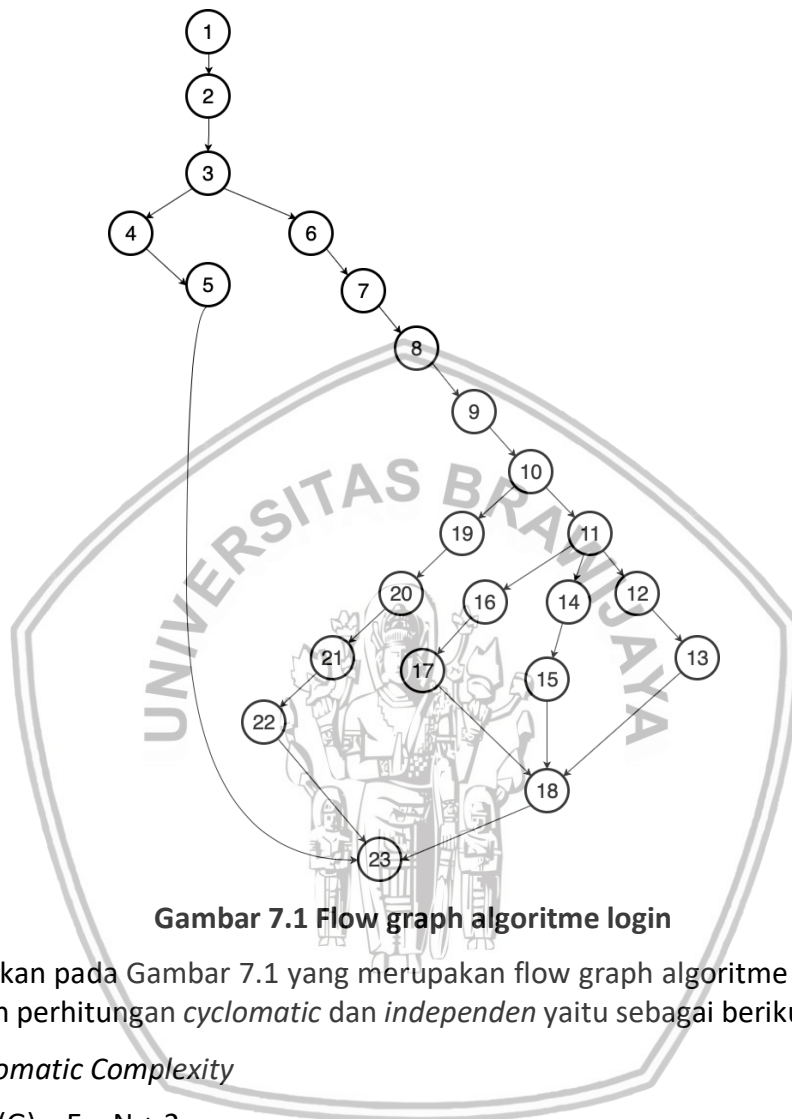

```

33     $log = $this->M_Index->create_log($data);
34     if ($log != 0) {
35         $session = array('id_user' => $id_user,
36             'nama' => $username,
37             'role' => $role);
38         $this->session->set_userdata($session);
39         redirect('C_Admin');
40     } break; 13
41
42 case 2: 14
43     $guru = $this->M_Guru->get_guru_by_id($id_user);
44     foreach ($guru as $key => $nilai) {
45         $nama = $nilai->nama_guru;
46     }
47     $update = $this->M_Index->update_login($last_login, $id_user);
48     $data = array('id_user' => $id_user,
49         'last_update' => $last_login,
50         'log_aktivitas' => $nama . '|Melakukan Login');
51     $log = $this->M_Index->create_log($data);
52     if ($log != 0) {
53         $session = array('id_user' => $id_user,
54             'nama' => $nama,
55             'role' => $role);
56         $this->session->set_userdata($session);
57         redirect('C_Guru');
58     } break; 15
59
60 case 3:// 16
61     $siswa = $this->M_Siswa->get_siswa_by_id($id_user);
62     foreach ($siswa as $key => $nilai) {
63         $nama = $nilai->nama;
64     }
65     $update = $this->M_Index->update_login($last_login, $id_user);
66     $data = array('id_user' => $id_user,
67         'last_update' => $last_login,
68         'log_aktivitas' => $nama . '|Melakukan Login');
69     $log = $this->M_Index->create_log($data);
70     if ($log != 0) {
71         $session = array('id_user' => $id_user,
72             'nama' => $nama,
73             'role' => $role);
74         $this->session->set_userdata($session);
75         redirect('C_Siswa');
76     } break; 17
77
78 default:
79     # code...
80     break; 18
81 } 19
82
83 else{ 20
84     $this->session->set_flashdata('flash_message', warn_msg('User tidak
85     ada'));
86     redirect('C_Index'); 21
87 } 22
88 } 23

```

Pada Tabel 7.1 dapat dilihat bawah dari algoritme *login*, terdapat angka 1 sampai dengan 23 dimana angka-angka tersebut menunjukkan banyaknya node yang akan diubah kedalam bentuk flow graph. Berikut ini dalam Gambar 7.1

merupakan hasil penggambaran flow graph yang didapat dari algoritme pada Tabel 7.1.



Gambar 7.1 Flow graph algoritme login

Berdasarkan pada Gambar 7.1 yang merupakan flow graph algoritme login dapat dilakukan perhitungan *cyclomatic* dan *independen* yaitu sebagai berikut

- *Cyclomatic Complexity*

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 26 - 23 + 2 \\ &= 5 \end{aligned}$$

$$\begin{aligned} V(G) &= P + 1 \\ &= 4 + 1 \\ &= 5 \end{aligned}$$

$$\begin{aligned} V(G) &= \text{Jumlah region} \\ &= 5 \end{aligned}$$

- *Independent path*

1. 1-2-3-4-5-23

2. 1-2-3-6-7-8-9-10-11-12-13-18-23
3. 1-2-3-6-7-8-9-10-11-14-15-18-23
4. 1-2-3-6-7-8-9-10-11-16-17-18-23
5. 1-2-3-6-7-8-9-10-19-20-21-22-23

Jalur independen yang telah terbentuk dapat dilakukan pengujian kasus uji pada masing-masing jalurnya. Pada Tabel 7.2 yang akan mendefinikan kasus uji pada setiap jalur independen yang ada.

Tabel 7.2 Kasus uji algoritme login

No	Jalur	Data Input	Hasil yang diharapkan	Hasil yang diperoleh	status
1.	1	Memanggil operasi login() dengan nilai form_validation = Salah.	Sistem menampilkan pesan <i>error</i> dan kembali pada halaman depan aplikasi <i>E-learning</i> .	Tampil halaman depan aplikasi <i>E-learning</i> dan pesan <i>error</i> pada jendela login.	Valid
2.	2	Memanggil operasi login() dengan nilai form_validation = Benar. Nilai dari variabel \$role adalah 1.	Sistem menampilkan halaman depan admin.	Tampil halaman depan admin.	Valid
3.	3	Memanggil operasi login() dengan nilai form_validation = Benar. Nilai dari variabel \$role adalah 2.	Sistem menampilkan halaman depan guru.	Tampil halaman depan guru.	Valid
4.	4	Memanggil operasi login() dengan nilai form_validation = Benar. Nilai dari variabel \$role adalah 3.	Sistem menampilkan halaman depan siswa.	Tampil halaman depan siswa.	Valid
5.	5	Memanggil operasi login() dengan nilai form_validation = Benar.	Sistem menampilkan halaman depan aplikasi <i>E-learning</i>	Tampil halaman depan aplikasi <i>E-learning</i> dan	Valid

		Nilai dari variabel \$hasil sama dengan kosong atau data user tidak ada.	dan pesan error "User tidak ada"/	tampil pesan error "User tidak ada" pada jendela login.	
--	--	--	-----------------------------------	---	--

7.1.2 Pengujian Unit Klas C_Guru

Operasi tambah_kelas()

Tabel 7.3 Algoritme tambah_kelas

1	function tambah_kelas(){
2	\$this->load->model('M_Index');
3	\$post = \$this->input->post();
4	\$date_now = date("Y-m-d H:i:s");
5	\$path .= './fl/class/';
6	\$_FILES['gambar']['name'] = \$_FILES['gambar']['name'];
7	\$_FILES['gambar']['type'] = \$_FILES['gambar']['type'];
8	\$_FILES['gambar']['tmp_name'] =
9	\$_FILES['gambar']['tmp_name'];
10	\$_FILES['gambar']['error'] = \$_FILES['gambar']['error'];
11	\$_FILES['gambar']['size'] = \$_FILES['gambar']['size'];
12	\$config['upload_path'] = \$path;
13	\$config['allowed_types'] = 'jpg png';
14	\$config['file_name'] = 'FC'.date("dmyHis");
15	\$this->load->library('upload', \$config);
16	if (\$this->upload->do_upload('gambar')) {
17	\$upload_data = \$this->upload->data();
18	\$gambar = \$path . \$upload_data['file_name'];
19	\$kelas = \$post["nama"] . " - " .
20	\$post["angkatan"];
21	\$foto = \$this->input->post('gambar');
22	\$data = array('id_matpel' => \$post['matpel'],
23	'nama_kelas' => \$kelas,
24	'enroll_key' => \$post['kode'],
25	'created_by' => \$post['id_guru'],
26	'created_at' => \$date_now,
27	'tahun_kelas' => \$post['angkatan'],
28	'keterangan' => \$post['keterangan'],
29	'gambar_kelas' => \$gambar);
30	\$tambah = \$this->M_Guru->tambah_kelas(\$data);
31	if (\$tambah != 0) {
32	\$datal=array('id_user'=>\$this->session->userdata('id_user'),
33	'last_update' => \$date_now,
34	'log_aktivitas'=>\$this->session->
35	>userdata('nama').' MembuatKelas '. \$kelas);
36	\$log = \$this->M_Index->create_log(\$datal);
37	redirect('C_Guru');
38	}
39	else{
40	\$this->session->set_flashdata('flash_message',
41	warn_msg('Gagal Membuat Kelas Baru'));
42	redirect('C_Guru');
43	}
44	}
45	else{

Pada Tabel 7.3 dapat dilihat bawah dari algoritme tambah_kelas(), terdapat angka 1 sampai dengan 14 dimana angka-angka tersebut menunjukkan banyaknya node yang akan diubah kedalam bentuk flow graph. Dalam Gambar 7.2 merupakan hasil penggambaran flow graph yang didapat dari algoritme pada Tabel 7.2.



- *Cyclomatic Complexity*

78

- *Independent path*
 1. 1-2-3-4-5-6-10-14
 2. 1-2-3-4-7-8-9-10-14
 3. 1-2-11-12-13-14

Jalur independen yang telah terbentuk dapat dilakukan pengujian kasus uji pada masing-masing jalurnya. Pada Tabel 7.4 yang akan mendefinikan kasus uji pada setiap jalur independen yang ada.

Tabel 7.4 Kasus uji algoritme tambah_tugas

No	Jalur	Data Input	Hasil yang diharapkan	Hasil yang diperoleh	status
1.	1	Memanggil operasi tambah_tugas(). nilai dari do_upload adalah "gambar". menambahkan data kedalam variabel \$tambah. nilai dari \$tambah tidak sama dengan 0. menambahkan data aktifitas kedalam variabel \$log.	Sistem menampilkan halaman depan guru.	Tampil halaman depan aguru.	Valid
2.	2	Memanggil operasi tambah_tugas(). nilai dari do_upload adalah "gambar". menambahkan data kedalam variabel \$tambah. nilai dari \$tambah sama dengan 0 .	Sistem menampilkan halaman tambah kelas dan menampilkan pesan <i>error</i> "gagal membuat kelas baru".	Tampil halaman tambah kelas dan menampilkan pesan <i>error</i> "gagal membuat kelas baru".	Valid
3.	3	Memanggil operasi tambah_tugas(). nilai dari	Sistem menampilkan halaman tambah kelas dan pesan	Tampil halaman tambah kelas dan pesan	Valid

		do_upload bukan "gambar".	error.	error " This is Warning ! you did not select a file to upload".	
--	--	---------------------------	--------	---	--

7.1.3 Pengujian Unit Klas C_Siswa

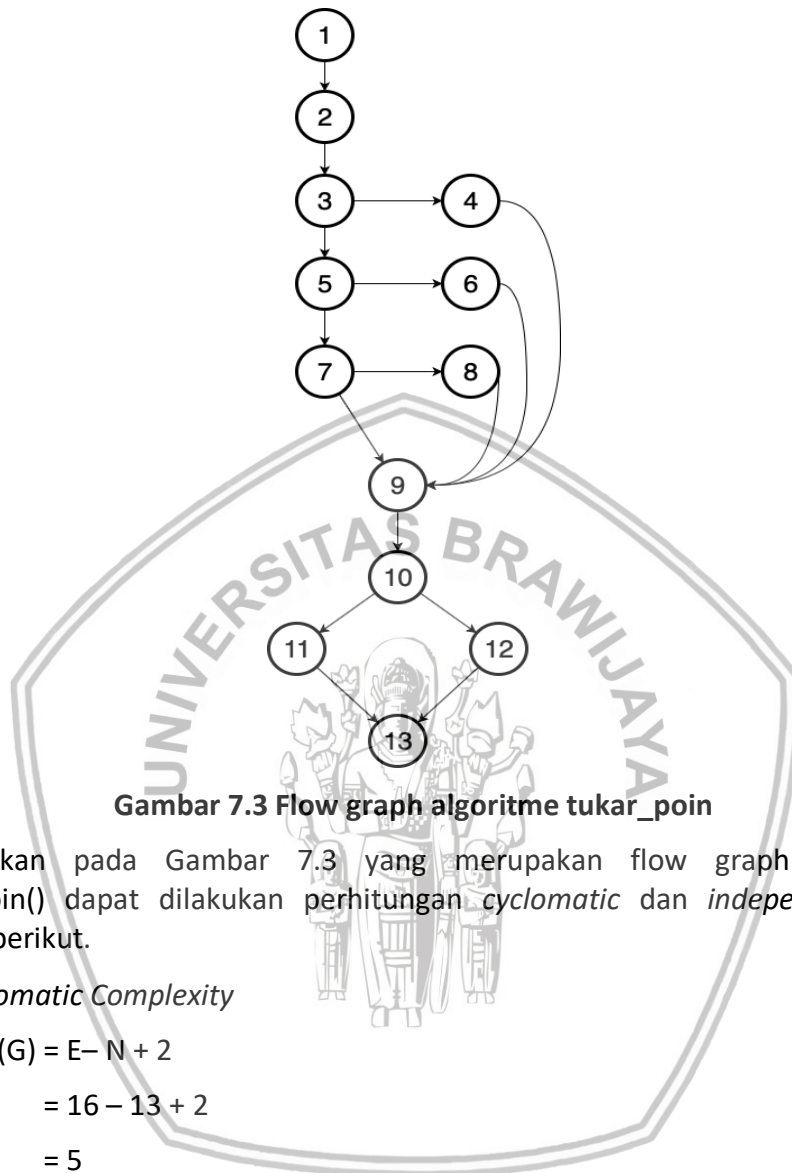
Operasi tukar_poin()

Tabel 7.5 Algoritme tukar_poin

1	function tukar_poin(){	
2	\$this->load->model('M_Index');	
3	\$post = \$this->input->post();	
4	\$last_update = date("Y-m-d H:i:s");	
5	\$jenis = '';	1
6	switch (\$post['jenis']) {	2
7	case 1:	3
8	\$jenis = 'Nilai Keaktifan';	
9	break;	4
10	case 2:	5
11	\$jenis = 'Nilai Kecerdasan';	
12	break;	6
13	case 3:	7
14	\$jenis = 'Nilai Sikap';	
15	break;	8
16	default:	
17	# code...	
18	break;	9
19	}	
20	if (\$post["jumlah_poin"] > \$post["total_poin"]) {	10
21	\$this->session->set_flashdata('flash_message',	
22	warn_msg('Poin Anda Tidak Dapat Ditukarkan, Karena Jumlah	
23	Poin Anda Kurang Untuk Penukaran Ini! Terima Kasih'));	11
24	redirect('C_Siswa/tukar');	
25	} else{	
26	\$data = array('id_siswa' => \$post["id_siswa"],	
27	'jenis_redeem'=> \$post["jenis"],	
28	'jumlah_poin' => \$post["jumlah_poin"],	12
29	'tanggal_redeem'=> \$last_update);	
30		
31	\$redeem = \$this->M_Siswa->create_redeem(\$data);	
32	\$data2=array('id_user'=>\$this->session-	
33	->userdata('id_user'),	
34	'last_update'=> \$last_update,	
35	'log_aktivitas'=>\$this->session-	12
36	->userdata('nama') . ' Menukarkan Poin	
37	Sebanyak ' . \$post["jumlah_poin"] . ' Ke	
38	Dalam ' . \$jenis);	
39	\$log = \$this->M_Index->create_log(\$data2);	
40	echo "<script language='JavaScript'>alert('Selamat anda	
	berhasil menukarkan poin	
);document.location='/elearning/C_Siswa/tukar'	
	</script>";	
	}	13

Pada Tabel 7.5 dapat dilihat bawah dari algoritme tukar_poin(), terdapat angka 1 sampai dengan 13 dimana angka-angka tersebut menunjukkan banyaknya node

yang akan diubah kedalam bentuk flow graph. Dalam gambar 7.3 merupakan hasil penggambaran flow graph yang didapat dari algoritme pada Tabel 7.3.



Gambar 7.3 Flow graph algoritme tukar_poin

Berdasarkan pada Gambar 7.3 yang merupakan flow graph algoritme tukar_poin() dapat dilakukan perhitungan *cyclomatic* dan *independen* yaitu sebagai berikut.

- *Cyclomatic Complexity*

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 16 - 13 + 2 \\
 &= 5
 \end{aligned}$$

$$\begin{aligned}
 V(G) &= P + 1 \\
 &= 4 + 1 \\
 &= 5
 \end{aligned}$$

$$\begin{aligned}
 V(G) &= \text{Jumlah region} \\
 &= 5
 \end{aligned}$$

- *Independent path*

1. 1-2-3-4-9-10-12-13
2. 1-2-3-5-6-9-10-12-13
3. 1-2-3-5-7-9-10-11-13

4. 1-2-3-5-7-9-10-11-13

5. 1-2-3-5-7-9-10-12-13

Jalur independen yang telah terbentuk dapat dilakukan pengujian kasus uji pada masing-masing jalurnya. Berikut ini merupakan Tabel 7.6 yang akan mendefinikan kasus uji pada setiap jalur independen yang ada.

Tabel 7.6 Kasus uji algoritme tukar poin

No	Jalur	Data Input	Hasil yang diharapkan	Hasil yang diperoleh	status
1.	1	Nilai dari switch = 1 dan jumlah poin yang dimasukkan lebih kecil dari total poin.	Sistem menampilkan pesan "selamat anda berhasil menukarkan poin" dan kembali kehalaman tukar poin.	Tampil pesan "selamat anda berhasil menukarkan poin" dan menampilkan halaman tukar poin	Valid
2.	2	Nilai dari switch = 2 dan jumlah poin yang dimasukkan lebih kecil dari total poin.	Sistem menampilkan pesan "selamat anda berhasil menukarkan poin" dan kembali kehalaman tukar poin.	Tampil pesan "selamat anda berhasil menukarkan poin" dan menampilkan halaman tukar poin.	Valid
3.	3	Nilai dari switch = 3 dan jumlah poin yang dimasukkan lebih kecil dari total poin.	Sistem menampilkan pesan "selamat anda berhasil menukarkan poin" dan kembali kehalaman tukar poin.	Tampil pesan "selamat anda berhasil menukarkan poin" dan menampilkan halaman tukar poin.	Valid
4.	4	Nilai dari switch != 1,2,3 dan jumlah poin yang dimasukkan lebih besar dari total poin.	Sistem menampilkan pesan <i>error</i> dan kembali pada halaman tukar poin.	tampil pesan <i>error</i> dan kembali ke halaman tukar poin.	Valid
5.	5	Nilai dari switch != 1,2,3 dan jumlah poin yang	Sistem menampilkan pesan <i>error</i> yaitu	Tampil jendela baru berisi pesan "gagal	Valid

		dimasukkan lebih kecil dari total poin.	"gagal menukarkan poin".	menukarkan poin".	
--	--	---	--------------------------	-------------------	--

7.2 Pengujian *Blackbox*

Pengujian *blackbox* merupakan langkah pengujian untuk melakukan validasi pada aplikasi yang telah diimplementasikan kedalam kode program. Pengujian *blackbox* yang dipilih dalam penelitian ini adalah jenis *requirement test*, *requirement test* memungkinkan setiap kebutuhan yang telah diimplementasikan kedalam kode program dapat diuji sesuai kasus uji yang normal serta kasus uji alternative. Skenario pengujian yang dilakukan adalah mencoba setiap fungsionalitas yang telah diimplementasikan pada sistem dengan menggunakan kasus uji normal dan kasus uji alternatif yang dibuat oleh penulis.

7.2.1 Pengujian *requirement login*

Tabel 7.7 Kasus uji *requirement login*

Kasus dan hasil uji (Kondisi normal)			
Prosedur Pengujian	Hasil yang diharapkan	Hasil pengamatan	Kesimpulan
Actor melakukan input <i>username</i> dan <i>password</i> dengan benar	Actor berhasil <i>login</i> kedalam sistem dan halaman depan aktor akan otomatis ditampilkan.	Jika <i>username</i> dan <i>password</i> yang inputkan siswa atau guru benar, maka aktor akan masuk kedalam sistem. Tepatnya berada dihalaman depan aktor	Valid
Kasus dan hasil uji(kondisi alternative)			
Jika pada saat <i>login</i> aktor memilih tombol <i>login</i> , <i>username</i> atau <i>password</i> kosong.	Sistem menampilkan pesan <i>error</i> pada salah satu kolom yang kosong, baik itu <i>password</i> atau <i>username</i> .	Terdapat pesan <i>error</i> yaitu " <i>fill out this field</i> " pada kolom yang kosong.	Valid
Jika pada saat <i>login</i> aktor memasukkan <i>username</i> atau <i>password</i> salah.	Sistem menampilkan pesan <i>error</i> atau pesan kesalahan.	Sistem menampilkan halaman depan aplikasi dan jika tombol <i>login</i> ditekan maka terdapat pesan pada jendela atas	Valid

		<i>form login</i> yaitu “ <i>This is warning! User tidak ada</i> ”.	
Jika pada saat <i>login</i> aktor memasukkan <i>username</i> salah.	Sistem menampilkan pesan <i>error</i> atau pesan kesalahan.	Sistem menampilkan halaman depan aplikasi dan jika tombol <i>login</i> ditekan maka terdapat pesan pada jendela atas <i>form login</i> yaitu “ <i>This is warning! User tidak ada</i> ”.	Valid
Jika pada saat <i>login</i> aktor memasukkan <i>password</i> salah.	Sistem menampilkan pesan <i>error</i> atau pesan kesalahan	Sistem menampilkan halaman depan aplikasi dan jika tombol <i>login</i> ditekan maka terdapat pesan pada jendela atas <i>form login</i> yaitu “ <i>This is warning! User tidak ada</i> ”.	Valid

7.2.2 Pengujian *requirement* register untuk siswa

Tabel 7.8 Kasus uji *requirement* register untuk siswa

Kasus dan hasil uji (Kondisi normal)			
Prosedur Pengujian	Hasil yang diharapkan	Hasil pengamatan	Kesimpulan
Siswa melakukan <i>input</i> data berupa <i>username</i> , <i>password</i> , nama, alamat, nomor hp, <i>email</i> , angkatan dan juga foto <i>profile</i> .	Siswa berhasil mendaftar dan data masukkan siswa tadi tersimpan didalam <i>database</i> .	Setelah siswa berhasil mendaftar dan semua data sudah tersimpan, sistem akan secara otomatis mengarahkan siswa kehalaman depan siswa.	Valid
Kasus dan hasil uji(kondisi alternative)			
Jika pada saat <i>register</i> siswa tidak mengisi salah satu atau	Sistem menampilkan pesan <i>error</i> pada salah satu kolom yang kosong.	Terdapat pesan <i>error</i> yaitu “ <i>fill out this field</i> ” pada kolom yang kosong.	Valid

semua <i>field</i> .			
----------------------	--	--	--

7.2.3 Pengujian *requirement register* untuk guru

Tabel 7.9 Kasus uji *requirement register* untuk guru

Kasus dan hasil uji (Kondisi normal)			
Prosedur Pengujian	Hasil yang diharapkan	Hasil pengamatan	Kesimpulan
Siswa melakukan <i>input</i> data berupa <i>username</i> , <i>password</i> , nama, alamat, nomor hp, <i>email</i> , dan juga foto <i>profile</i> dengan benar.	Guru berhasil mendaftar dan data masukkan siswa tadi tersimpan didalam <i>database</i> .	Setelah guru berhasil mendaftar dan semua data sudah tersimpan, sistem akan secara otomatis mengarahkan guru kehalaman depan guru.	Valid
Kasus dan hasil uji(kondisi alternative)			
Jika pada saat <i>register</i> guru tidak mengisi salah satu atau semua <i>field</i> .	Sistem menampilkan pesan <i>error</i> pada salah satu kolom yang kosong.	Terdapat pesan <i>error</i> yaitu “ <i>fill out this field</i> ” pada kolom yang kosong.	Valid

7.2.4 Pengujian *requirement* membuat soal

Tabel 7.10 Kasus uji *requirement* membuat soal

Kasus dan hasil uji (Kondisi normal)			
Prosedur Pengujian	Hasil yang diharapkan	Hasil pengamatan	Kesimpulan
Guru memasukkan data kelas, bab, soal, jawaban dengan benar.	Sistem berhasil membuat soal dan menyimpan data tersebut dalam <i>datbase</i>	Setelah sistem berhasil membuat soal, maka sistem langsung secara otomatis kembali kehalaman buat soal.	Valid
Kasus dan hasil uji(kondisi alternative)			
Jika pada saat membuat soal guru tidak memasukkan data soal	Sistem menampilkan pesan <i>error</i> pada kolom soal. Yaitu “fill out this field”.	Terdapat pesan <i>error</i> yaitu “ <i>fill out this field</i> ” pada kolom soal.	Valid

Jika pada saat membuat soal guru tidak memasukkan data pilihan jawaban	Sistem menampilkan pesan <i>error</i> pada kolom jawaban. Yaitu "fill out this field".	Terdapat pesan <i>error</i> yaitu " fill out this field" pada kolom jawaban.	Valid
Jika guru memilih jenis soal pilihan ganda.	Sistem menampilkan halaman soal dalam bentuk pilihan ganda.	Sitem menampilkan halaman pilihan ganda.	Valid
Jika guru memilih jenis soal <i>drag & dorp</i> .	Sistem menampilkan halaman soal dalam bentuk <i>drag & dorp</i> .	Sitem menampilkan halaman <i>drag & dorp</i> .	Valid
Jika guru memilih jenis soal acak kata.	Sistem menampilkan halaman soal dalam bentuk acak kata.	Sitem menampilkan halaman acak kata.	Valid
Jika guru memilih jenis soal pilhan ganda gambar.	Sistem menampilkan halaman soal dalam bentuk pilhan ganda gambar.	Sitem menampilkan halaman pilhan ganda gambar.	Valid

7.2.5 Pengujian *requirement* membuat kelas

Tabel 7.11 Kasus uji *requirement* membuat kelas

Kasus dan hasil uji (Kondisi normal)			
Prosedur Pengujian	Hasil yang diharapkan	Hasil pengamatan	Kesimpulan
Guru memasukkan data kelas berupa nama kelas, mata pelajaran, angkatan, keterangan kelas, <i>enroll code</i> dengan benar.	Sistem berhasil menyimpan data kelas kedalam data base, dan berhasil membuat kelas baru.	Setelah sistem berhasil membuat kelas baru, secara otomatis sistem menampilkan halaman depan guru yang berisi daftar kelas.	Valid
Kasus dan hasil uji(kondisi alternative)			
Jika pada saat akan membuat kelas guru memilih tombol batal	Sistem menampilkan halaman daftar kelas atau halaman depan guru.	Sistem menampilkan halaman depan guru yang berisi daftar kelas.	Valid

7.2.6 Pengujian *requirement* melihat daftar siswa (guru)

Tabel 7.12 Kasus uji *requirement* melihat daftar siswa (guru)

Kasus dan hasil uji (Kondisi normal)			
Prosedur Pengujian	Hasil yang diharapkan	Hasil pengamatan	Kesimpulan
Guru memilih tombol lihat siswa yang ada pada daftar kelas.	Sistem berhasil menampilkan halaman daftar siswa dalam bentuk tabel.	Sistem menampilkan daftar siswa dalam halaman daftar siswa dalam bentuk tabel yang berisi no, nama, nomor hp, <i>email</i> , angkatan, tugas, <i>last update</i> , dan <i>icon</i> untuk view <i>profile</i> .	Valid

7.2.7 Pengujian *requirement* melihat *profile* siswa

Tabel 7.13 Kasus uji *requirement* melihat *profile* siswa

Kasus dan hasil uji (Kondisi normal)			
Prosedur Pengujian	Hasil yang diharapkan	Hasil pengamatan	Kesimpulan
Guru memilih <i>icon</i> pada siswa tertentu yang ada di daftar tabel.	Sistem berhasil menampilkan halaman profil siswa.	Sistem menampilkan daftar siswa dalam halaman profil siswa yang berisi nama, foto <i>profile</i> , alamat, nomor hp, <i>email</i> , angkatan, <i>log</i> aktifitas yang berisi nama, tanggal log. Dan juga data statistik yang berisi jumlah poin, tabel dengan data mata pelajaran, bab, nilai..	Valid

7.2.8 Pengujian *requirement* upload materi

Tabel 7.14 Kasus uji *requirement* upload materi

Kasus dan hasil uji (Kondisi normal)			
Prosedur Pengujian	Hasil yang diharapkan	Hasil pengamatan	Kesimpulan

Guru memasukkan data <i>upload</i> yang berisi nama materi, dan memilih <i>choose file</i> .	Sistem berhasil menyimpan data <i>upload</i> kedalam database.	Setelah sistem berhasil menyimpan data <i>upload</i> kedalam database maka secara otomatis sistem menampilkan halaman daftar bab.	Valid
Kasus dan hasil uji(kondisi alternative)			
Jika guru tidak mengisikan nama materi	Sistem menampilkan pesan <i>error</i> yang berisi " <i>Please fill out this field</i> ".	Sistem menampilkan pesan <i>error</i> pada kolom materi yang berisi " <i>Please fill out this field</i> ".	Valid

7.2.9 Pengujian *requirement* menghapus soal

Tabel 7.15 Kasus uji *requirement* menghapus soal

Kasus dan hasil uji (Kondisi normal)			
Prosedur Pengujian	Hasil yang diharapkan	Hasil pengamatan	Kesimpulan
Guru memilih tombol <i>icon</i> sampah pada daftar soal yang ada. Lalu memilih tombol <i>ok</i> pada pesan yang muncul.	Sistem menampilkan pesan konfirmasi, dan berhasil menghapus data soal dari <i>database</i> .	Setelah sistem berhasil menghapus data soal dari <i>database</i> , maka secara otomatis sistem menampilkan halaman daftar soal.	Valid
Kasus dan hasil uji(kondisi alternative)			
Jika guru memilih tombol <i>cancel</i> pada pesan konfirmasi yang muncul.	Sistem menampilkan halaman daftar soal dan tidak menghapus data soal dari <i>database</i> .	Sistem menampilkan halaman daftar soal.	Valid

7.2.10 Pengujian *requirement* memperbarui soal

Tabel 7.16 Kasus uji *requirement* memperbarui soal

Kasus dan hasil uji (Kondisi normal)			
Prosedur	Hasil yang	Hasil pengamatan	Kesimpulan

Pengujian	diharapkan		
Guru memilih tombol <i>icon</i> edit pada daftar soal yang ada.	Sistem menampilkan halaman soal yang berisi soal, jawaban, jawaban benar, dan tombol ubah soal.	Sistem menampilkan halaman soal yang berisi soal, jawaban, jawaban benar, dan tombol ubah soal.	Valid
Kasus dan hasil uji(kondisi alternative)			
Jika guru tidak melakukan perubahan pada data soal lalu memilih tombol ubah soal.	Sistem menampilkan halaman soal tersebut dan tidak ada perubahan data soal.	Sistem menampilkan halaman soal dengan data yang sama.	Valid

7.2.11 Pengujian *requirement* melihat daftar soal

Tabel 7.17 Kasus uji *requirement* melihat daftar soal

Kasus dan hasil uji (Kondisi normal)			
Prosedur Pengujian	Hasil yang diharapkan	Hasil pengamatan	Kesimpulan
Guru memilih menu daftar soal	Sistem berhasil menampilkan halaman daftar soal.	Sistem menampilkan halaman daftar soal.	Valid

7.2.12 Pengujian *requirement* *upload* tugas (guru)

Tabel 7.18 Kasus uji *requirement* *upload* tugas (guru)

Kasus dan hasil uji (Kondisi normal)			
Prosedur Pengujian	Hasil yang diharapkan	Hasil pengamatan	Kesimpulan
Guru memasukkan data <i>upload</i> yang berisi nama tugas, dan memilih <i>choose file</i> .	Sistem berhasil menyimpan data tugas kedalam database.	Setelah sistem berhasil menyimpan data tugas kedalam database maka secara otomatis sistem menampilkan halaman daftar bab.	Valid
Kasus dan hasil uji(kondisi alternative)			
Jika guru tidak mengisikan nama tugas	Sistem menampilkan pesan <i>error</i> yang berisi " <i>Please fill out</i>	Sistem menampilkan pesan <i>error</i> pada kolom tugas yang	Valid

	<i>this field</i> ” pada kolom yang kosong.	berisi “ <i>Please fill out this field</i> ”.	
--	---	---	--

7.2.13 Pengujian *requirement* keluar

Tabel 7.19 Kasus uji *requirement* keluar

Kasus dan hasil uji (Kondisi normal)			
Prosedur Pengujian	Hasil yang diharapkan	Hasil pengamatan	Kesimpulan
Aktor memilih tombol keluar.	Sistem berhasil menghapus data <i>session login</i> dari aktor dan menampilkan halaman depan aplikasi <i>E-learning</i> .	Sistem berhasil menampilkan halaman depan aplikasi <i>E-learning</i> .	Valid

7.2.14 Pengujian *requirement* melihat daftar guru

Tabel 7.20 Kasus uji *requirement* melihat daftar guru

Kasus dan hasil uji (Kondisi normal)			
Prosedur Pengujian	Hasil yang diharapkan	Hasil pengamatan	Kesimpulan
Admin memilih menu daftar guru.	Sistem berhasil menampilkan halaman daftar guru dalam bentuk tabel yang berisi no, nama, alamat, nomor hp, <i>email</i> , dan juga aksi.	Sistem menampilkan halaman daftar guru dalam bentuk tabel yang berisi no, nama, alamat, nomor hp, <i>email</i> , dan juga aksi.	Valid

7.2.15 Pengujian *requirement* melihat daftar siswa(admin)

Tabel 7.21 Kasus uji *requirement* melihat daftar siswa (admin)

Kasus dan hasil uji (Kondisi normal)			
Prosedur Pengujian	Hasil yang diharapkan	Hasil pengamatan	Kesimpulan
Admin memilih menu daftar siswa.	Sistem berhasil menampilkan halaman daftar siswa dalam bentuk tabel	Sistem menampilkan halaman daftar siswa dalam bentuk tabel yang berisi no, nama,	Valid

	yang berisi no, nama, alamat , nomor hp, <i>email</i> , angkatan dan juga aksi.	alamat , nomor hp, <i>email</i> , angkatan dan juga aksi.	
--	---	---	--

7.2.16 Pengujian *requirement* melihat daftar aktifitas

Tabel 7.22 Kasus uji *requirement* melihat daftar aktifitas

Kasus dan hasil uji (Kondisi normal)			
Prosedur Pengujian	Hasil yang diharapkan	Hasil pengamatan	Kesimpulan
Admin memilih menu daftar aktifitas.	Sistem berhasil menampilkan halaman daftar aktifitas dalam bentuk tabel yang berisi no, nama, tanggal , <i>log</i> .	Sistem berhasil menampilkan halaman daftar aktifitas dalam bentuk tabel yang berisi no, nama, tanggal , <i>log</i> .	Valid

7.2.17 Pengujian *requirement* menghapus akun guru

Tabel 7.23 Kasus uji *requirement* menghapus akun guru

Kasus dan hasil uji (Kondisi normal)			
Prosedur Pengujian	Hasil yang diharapkan	Hasil pengamatan	Kesimpulan
Admin memilih tombol hapus akun pada salah satu data akun guru yang ada di tabel.	Sistem berhasil menghapus data guru dari <i>database</i> dan menampilkan halaman daftar guru.	Sistem berhasil menampilkan halaman daftar guru, dan akun yang dihapus tidak muncul pada daftar guru.	Valid

7.2.18 Pengujian *requirement* menghapus akun siswa

Tabel 7.24 Kasus uji *requirement* menghapus akun siswa

Kasus dan hasil uji (Kondisi normal)			
Prosedur Pengujian	Hasil yang diharapkan	Hasil pengamatan	Kesimpulan
Admin memilih tombol hapus akun pada salah	Sistem berhasil menghapus data siswa dari <i>database</i>	Sistem berhasil menampilkan halaman daftar siswa,	Valid

satu data akun siswa yang ada di tabel.	dan menampilkan halaman daftar siswa.	dan akun yang dihapus tidak muncul pada daftar siswa.	
---	---------------------------------------	---	--

7.2.19 Pengujian *requirement* melakukan *competition*

Tabel 7.25 Kasus uji *requirement* melakukan *competition*

Kasus dan hasil uji (Kondisi normal)			
Prosedur Pengujian	Hasil yang diharapkan	Hasil pengamatan	Kesimpulan
Siswa memilih kelas dan siswa yang ingin ditantang, kemudian memilih tombol tantang.	Sistem berhasil mengirimkan <i>request</i> pada siswa yang ditantang dan juga menampilkan pesan “anda telah berhasil menantang nama siswa, silahkan tunggu <i>response</i> selanjutnya”.	Sistem menampilkan pesan “anda telah berhasil menantang nama siswa, silahkan tunggu <i>response</i> selanjutnya”.	Valid
Kasus dan hasil uji(kondisi alternative)			
Jika siswa memilih tombol batal pada saat berada dihalaman <i>competition</i> .	Sistem menampilkan halaman depan siswa.	Sistem menampilkan halaman depan siswa.	Valid

7.2.20 Pengujian *requirement* melakukan *challenge*

Tabel 7.26 Kasus uji *requirement* melakukan *challenge*

Kasus dan hasil uji (Kondisi normal)			
Prosedur Pengujian	Hasil yang diharapkan	Hasil pengamatan	Kesimpulan
Siswa memilih tombol mulai <i>challenge</i> . Dan memilih tombol <i>close</i> pada pesan konfirmasi.	Sistem menampilkan pesan konfirmasi dan menampilkan halaman soal <i>challenge</i> .	Sistem berhasil menampilkan pesan konfirmasi dan juga menampilkan soal yang akan dikerjakan oleh siswa.	Valid
Siswa mengerjakan	Sistem akan	Sistem menampilkan	Valid

semua soal dari awal hingga akhir lalu memilih tombol submit.	menampilkan pesan konfirmasi dan setelah memilih tombol <i>close</i> sistem menampilkan halaman depan siswa.	pesan konfirmasi dan akan menampilkan halaman depan siswa jika siswa memilih tombol <i>close</i> pada pesan konfirmasi tersebut.	
Kasus dan hasil uji(kondisi alternative)			
Jika pada saat setelah mengerjakan <i>challenge</i> mendapatkan poin sama dengan atau diatas 80.	Sistem menampilkan pesan pesan “Sistem menampilkan pesan “ selamat anda berhasil mengerjakan soal dengan jumlah (jumlah soal) dan Mendapatkan Nilai : (nilai) , dan poin kamu telah mendapatkan tambhan poin”.	Sistem menampilkan pesan pesan “Sistem menampilkan pesan “ selamat anda berhasil mengerjakan soal dengan jumlah (jumlah soal) dan Mendapatkan Nilai : (nilai) , dan poin kamu telah mendapatkan tambhan poin”.	Valid
Jika pada saat setelah mengerjakan <i>challenge</i> mendapatkan poin dibawah 80.	Sistem menampilkan pesan “ selamat anda berhasil mengerjakan soal dengan jumlah (jumlah soal) dan Mendapatkan Nilai : (nilai) , Namun Anda Tidak Mendapatkan Tambahan Poin Karena Nilai Anda Kurang Dari 80”.	Sistem menampilkan pesan “ selamat anda berhasil mengerjakan soal dengan jumlah (jumlah soal) dan Mendapatkan Nilai : (nilai) , Namun Anda Tidak Mendapatkan Tambahan Poin Karena Nilai Anda Kurang Dari 80”.	Valid

7.2.21 Pengujian *requirement* melihat peringkat

Tabel 7.27 Kasus uji *requirement* melihat peringkat

Kasus dan hasil uji (Kondisi normal)			
Prosedur Pengujian	Hasil yang diharapkan	Hasil pengamatan	Kesimpulan
Siswa memilih menu peringkat	Sistem berhasil menampilkan	Sistem menampilkan halaman peringkat	Valid

	halaman peringkat yang berisi data siswa dalam bentuk tabel dengan data, nama, kelas, angkatan, dan juga jumlah poin.	yang berisi data siswa dalam bentuk tabel dengan data, nama, kelas, angkatan, dan juga jumlah poin.	
Kasus dan hasil uji(kondisi alternative)			
Jika siswa memilih menu kelas pada tabel peringkat.	sistem akan menampilkan tabel yang disusun berdasarkan kelas yang dipilih.	Sistem menampilkan tabel yang disusun berdasarkan kelas.	Valid
Jika siswa memilih menu angkatan pada tabel peringkat.	sistem akan menampilkan tabel yang disusun berdasarkan angkatan yang dipilih.	Sistem menampilkan tabel yang disusun berdasarkan angkatan.	Valid

7.2.22 Pengujian *requirement* melihat *profile*

Tabel 7.28 Kasus uji *requirement* melihat *profile*

Kasus dan hasil uji (Kondisi normal)			
Prosedur Pengujian	Hasil yang diharapkan	Hasil pengamatan	Kesimpulan
Siswa memilih menu <i>profile</i> .	Sistem berhasil menampilkan halaman <i>profile</i> siswa yang berisi nama, alamat, nomor hp, <i>email</i> , angkatan, foto profile jumlah poin, dan juga data statistik yang dalam bentuk tabel dan berisi no, mata pelajaran, bab, nilai.	Sistem menampilkan halaman <i>profile</i> siswa yang berisi nama, alamat, nomor hp, <i>email</i> , angkatan, foto profile jumlah poin, dan juga data statistik yang dalam bentuk tabel dan berisi no, mata pelajaran, bab, nilai.	Valid

7.2.23 Pengujian *requirement upload* tugas (siswa)

Tabel 7.29 Kasus uji *requirement upload* tugas (siswa)

Kasus dan hasil uji (Kondisi normal)			
Prosedur Pengujian	Hasil yang diharapkan	Hasil pengamatan	Kesimpulan
Siswa memilih tombol <i>upload</i> tugas.	Sistem berhasil menampilkan jendela baru yang berisi nama tugas dan juga tombol <i>choose file</i> .	Sistem menampilkan jendela baru yang berisi nama tugas dan juga tombol <i>choose file</i> .	Valid
Siswa memasukkan data tugas dan memilih tombol <i>upload</i> tugas.	Sistem berhasil menyimpan data tugas didalam <i>database</i> .	Sistem menampilkan halaman bab.	Valid
Kasus dan hasil uji(kondisi alternative)			
Jika siswa tidak mengisi nama tugas.	Sistem akan menampilkan pesan <i>error</i> yaitu "Please fill out this field".	Sistem menampilkan pesan <i>error</i> yaitu "Please fill out this field".	Valid

7.2.24 Pengujian *requirement* tambah kelas

Tabel 7.30 Kasus uji *requirement* tambah kelas

Kasus dan hasil uji (Kondisi normal)			
Prosedur Pengujian	Hasil yang diharapkan	Hasil pengamatan	Kesimpulan
Siswa memilih kelas, memasukkan <i>enroll code</i> , dan memilih tombol buat.	Sistem akan menampilkan pesan konfirmasi "anda yakin ingin bergabung dengan kelas ini?".	Sistem menampilkan pesan konfirmasi "anda yakin ingin bergabung dengan kelas ini?".	Valid
Siswa memilih tombol ok	Sistem akan menampilkan pesan konfirmasi berupa "selamat anda berhasil	Sistem menampilkan pesan konfirmasi berupa "selamat anda berhasil menambahkan	Valid

	menambahkan kelas".	kelas".	
Siswa memilih tombol <i>close</i> .	Sistem akan menambahkan siswa tersebut ke kedalam kelas yang dipilih, dan sistem menampilkan halaman depan siswa.	Sistem menampilkan halaman depan siswa.	Valid
Kasus dan hasil uji(kondisi alternative)			
Jika siswa memilih tombol batal	Sistem akan menampilkan halaman depan siswa.	Sistem menampilkan halaman depan siswa.	Valid

7.2.25 Pengujian *requirement* menurunkan poin

Tabel 7.31 Kasus uji *requirement* menukarkan poin

Kasus dan hasil uji (Kondisi normal)			
Prosedur Pengujian	Hasil yang diharapkan	Hasil pengamatan	Kesimpulan
Siswa memilih jenis penukaran poin, memasukkan jumlah poin, dan memilih tombol tukar poin.	Sistem akan menampilkan pesan konfirmasi"anda yakin ingin menukarkan poin".	Sistem menampilkan pesan konfirmasi"anda yakin ingin menukarkan poin".	Valid
Siswa memilih tombol ok	Sistem akan mengurangi jumlah poin siswa dan menampilkan halaman tukar poin.	Sistem menampilkan halaman tukar poin dan jumlah poin berubah.	Valid
Kasus dan hasil uji(kondisi alternative)			
Jika siswa tidak memiliki poin yang cukup saat memilih tombol tukar poin.	Sistem menampilkan pesan "this is a warning! Poin anda tidak dapat ditukarkan, karena jumlah poin anda kurang untuk penukaran ini! Terima kasih".	Sistem menampilkan "this is a warning! Poin anda tidak dapat ditukarkan, karena jumlah poin anda kurang untuk penukaran ini! Terima kasih".	Valid

7.2.26 Pengujian *requirement* melakukan *quiz*

Tabel 7.32 Kasus uji *requirement* melakukan *quiz*

Kasus dan hasil uji (Kondisi normal)			
Prosedur Pengujian	Hasil yang diharapkan	Hasil pengamatan	Kesimpulan
Siswa memilih tombol mulai <i>quiz</i> . Dan memilih tombol <i>close</i> pada pesan konfirmasi.	Sistem menampilkan pesan konfirmasi dan menampilkan halaman soal <i>quiz</i> .	Sistem berhasil menampilkan pesan konfirmasi dan juga menampilkan soal yang akan dikerjakan oleh siswa.	Valid
Siswa mengerjakan semua soal dari awal hingga akhir lalu memilih tombol submit.	Sistem akan menampilkan pesan konfirmasi dan setelah memilih tombol <i>close</i> sistem menampilkan halaman depan siswa.	Sistem menampilkan pesan konfirmasi dan akan menampilkan halaman depan siswa jika siswa memilih tombol <i>close</i> pada pesan konfirmasi tersebut.	Valid
Kasus dan hasil uji(kondisi alternative)			
Jika pada saat setelah mengerjakan <i>quiz</i> mendapatkan poin sama dengan atau diatas 80.	Sistem menampilkan pesan pesan "Sistem menampilkan pesan " selamat anda berhasil mengerjakan soal dengan jumlah (jumlah soal) dan Mendapatkan Nilai : (nilai) , dan poin kamu telah mendapatkan tambhan poin".	Sistem menampilkan pesan pesan "Sistem menampilkan pesan " selamat anda berhasil mengerjakan soal dengan jumlah (jumlah soal) dan Mendapatkan Nilai : (nilai) , dan poin kamu telah mendapatkan tambhan poin".	Valid
Jika pada saat setelah mengerjakan <i>quiz</i> mendapatkan poin dibawah 80.	Sistem menampilkan pesan " selamat anda berhasil mengerjakan soal dengan jumlah (jumlah soal) dan Mendapatkan Nilai : (nilai) , Namun Anda Tidak Mendapatkan	Sistem menampilkan pesan " selamat anda berhasil mengerjakan soal dengan jumlah (jumlah soal) dan Mendapatkan Nilai : (nilai) , Namun Anda Tidak Mendapatkan Tambahan Poin	Valid

	Tambahan Poin Karena Nilai Anda Kurang Dari 80".	Karena Nilai Anda Kurang Dari 80".	
--	--	---------------------------------------	--

7.2.27 Pengujian *requirement* melihat *about*

Tabel 7.33 Kasus uji *requirement* melihat *about*

Kasus dan hasil uji (Kondisi normal)			
Prosedur Pengujian	Hasil yang diharapkan	Hasil pengamatan	Kesimpulan
User memilih menu <i>about</i> pada halaman depan aplikasi	Sistem menampilkan halaman <i>about</i>	Sistem berhasil menampilkan halaman <i>about</i>	Valid

7.2.28 Pengujian *requirement* melihat *contact*

Tabel 7.34 Kasus uji *requirement* melihat *contact*

Kasus dan hasil uji (Kondisi normal)			
Prosedur Pengujian	Hasil yang diharapkan	Hasil pengamatan	Kesimpulan
User memilih menu <i>contact</i> pada halaman depan aplikasi	Sistem menampilkan halaman <i>contact</i>	Sistem berhasil menampilkan halaman <i>contact</i>	Valid

7.3 User Engagement Scale

Pada Tabel 7.35 adalah tabel hasil kuesioner *user engagement scale* (UES) beserta dengan hasil Min, Max, Mean, dan juga standar deviasi. Tabel dibawah ini adalah hasil dari perhitungannya.

Tabel 7.35 Tabel pengolahan data *user engagement scale*

Pertanyaan	Peserta Eksperimen			MIN	MAX	MEAN	Standar Deviasi
	P1	P2	P3				
Q1	4	3	4	3	4	3,67	0,58
Q2	4	3	4	3	4	3,67	0,58
Q3	3	2	2	2	3	2,33	0,58
Q4	2	4	2	2	4	2,67	1,15

Q5	4	4	2	2	4	3,33	1,15
Q6	4	4	4	4	4	4	0
Q7	5	5	4	4	5	4,67	0,58
Q8	2	2	2	2	2	2	0
Q9	2	2	2	2	2	2	0
Q10	2	2	2	2	2	2	0
Q11	2	2	2	2	2	2	0
Q12	2	2	2	2	2	2	0
Q13	2	2	1	1	2	1,67	0,58
Q14	4	5	4	4	5	4,33	0,58
Q15	3	2	2	2	3	2,33	0,58
Q16	4	5	5	4	5	4,67	0,58
Q17	4	5	5	4	5	4,67	0,58
Q18	4	5	5	4	5	4,67	0,58
Q19	4	5	5	4	5	4,67	0,58
Q20	4	4	5	4	5	4,33	0,58
Q21	4	4	5	4	5	4,33	0,58
Q22	4	4	4	4	4	4	0
Q23	2	3	2	2	3	2,33	0,58
Q24	4	4	5	4	5	4,33	0,58
Q25	3	5	5	3	5	4,33	1,15
Q26	3	4	4	3	4	3,67	0,58
Q27	4	4	4	4	4	4	0
Q28	4	3	5	3	5	4	1
Q29	4	4	4	4	4	4	0
Q30	4	5	5	4	5	4,67	0,58

Dari data yang ada pada table diatas, dapat dilihat bahwa hasil mean tertinggi adalah 4,67. Untuk nilai minimum yang dipilih peserta eksperimen adalah 1 dan untuk nilai maksimalnya adalah 5. Dari 30 item pernyataan yang ada, ada 26 pernyataan yang memiliki nilai standar deviasi dibawah 1. Hal ini menunjukkan bahwa tidak adanya perbedaan yang signifikan ketika para peserta eksperimen mengisi kuesioner UES, karena jika standar deviasi memiliki nilai di bawah 1 (0,55-0,89) berarti bahwa variannya sedikit, sedangkan jika standar deviasi memiliki nilai di atas 1 (1-1,48) berarti bahwa variannya banyak.

Pada tabel tabulasi hasil kuesioner UES dapat dilihat bahwa pernyataan dengan nomor 16,17,18,19 dan 30 merupakan pernyataan dengan hasil mean tertinggi yaitu 4,67 dan memiliki standar deviasi 0,58. Terlihat bahwa adanya korelasi antara hasil mean dan standar deviasi yang menunjukkan bahwa penyebaran, keanekaragaman dan varian hasil pemilihan jawaban peserta eksperimen menunjukkan perbedaan yang signifikan. Hal ini bisa disebabkan karena adanya perbedaan persepsi yang dirasakan oleh peserta eksperimen terkait dengan interaksinya pada aplikasi *E-learning*. Berbeda dengan pernyataan nomor 6, 8-12, 22, 27, 29, dimana hasil meannya adalah 2 dan 4 dimana standar deviasinya bernilai 0 yang menunjukkan adanya korelasi antara hasil mean dan standar deviasi bahwa penyebaran, keanekaragaman dan varian hasil pemilihan jawaban peserta eksperimen menunjukkan tidak adanya perbedaan yang signifikan (semua peserta eksperimen memilih jawaban yang sama). Hal ini bisa disebabkan karena tidak ada perbedaan persepsi yang dirasakan oleh peserta eksperimen terkait dengan interaksinya pada aplikasi *E-learning*. Pernyataan nomor 27 berbunyi “konten dari aplikasi *E-learning* menarik perhatian saya” dengan hasil seperti itu dapat dikatakan bahwa menurut persepsi para peserta eksperimen konten yang ada pada aplikasi *E-learning* ini 50:50 dapat menarik perhatian mereka.

Untuk keseluruhan hasil dari kuesioner UES dapat dikatakan bahwa para peserta eksperimen merasakan hal yang sama ketika menggunakan aplikasi *E-learning*, namun ada kalanya juga para peserta eksperimen memiliki perbedaan persepsi akan pengalaman emosional, kognitif dan perilaku peserta eksperimen ketika menggunakan aplikasi *E-learning* yang ditunjukkan dengan pemilihan jawaban kuesioner UES.

Tabel 7.36 Tabel skor *user engagement scale*

No	Item	Skor
1	<i>Focused attention</i>	3,46
2	<i>Perceived usability</i>	2,3
3	<i>Aesthetic appeal</i>	4,6
4	<i>Reward Factor</i>	3,96

Pada Tabel 7.36 merupakan hasil rata-rata skor kuesioner yang sudah diisi oleh tiga responden tersebut.

7.4 System Usability Scale

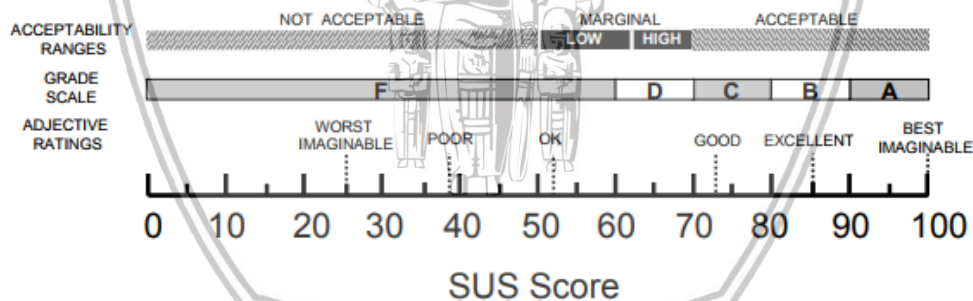
Perhitungan skor dari kuesioner SUS yaitu untuk mengetahui apakah skor yang didapat dapat memenuhi atau tidak memenuhi persyaratan untuk sistem itu diterima dan digunakan. Dibawah ini adalah perhitungan skor SUS dimana Q

melambangkan pertanyaan dan R adalah responden. Contoh perhitungan untuk skor SUS.

$$\begin{aligned}
 \text{Skor} &= (Q1 - 1) + (5 - Q2) + (Q3 - 1) + (5 - Q4) + (Q5 - 1) + (5 - Q6) \\
 &\quad + (Q7 - 1) + (5 - Q8) + (Q9 - 1) + (5 - Q10) \\
 &= (3 - 1) + (5 - 4) + (5 - 1) + (5 - 2) + (4 - 1) + (5 - 2) + (4 - 1) \\
 &\quad + (5 - 1) + (4 - 1) + (5 - 4) \\
 &= 27 \times 2.5 \\
 &= 67.5
 \end{aligned}$$

Tabel 7.37 Tabel skor *system usability scale*

Responden (R)	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Skor (SUS)
R1	3	4	5	2	4	2	4	1	4	4	67.5
R2	4	3	5	2	5	3	5	2	3	4	70
R3	4	2	4	4	4	2	4	2	4	4	65
Rata-rata skor											67.5



Gambar 7.4 SUS skor

Berdasarkan perhitungan skor kuesioner pada Tabel 7.37, rata-rata skor yang didapatkan dari 3 orang responden adalah 67,5. Jika dilihat dalam Gambar 7.4, angka 67,5 menunjukkan bahwa sistem yang ada di aplikasi *e-learning* ada di rentang huruf D atau *ACCEPTABLE HIGH*, itu artinya aplikasi *E-learning* termasuk dalam kategori layak dan dapat diterima oleh pengguna (*ACCEPTABLE*). Adanya beberapa responden yang tidak mengerti banyak tentang bagaimana menggunakan sebuah aplikasi sangat mempengaruhi nilai dari skor yang ada diatas, seperti pada kasus penelitian ini guru yang dijadikan sebagai objek pengujian untuk dapat memenuhi data sangat jarang mengakses sistem-sistem pembelajaran berbasis *online* seperti ini, sehingga dalam penggunaanya harus di arahkan dan nantinya akan berdampak pada kuesioner yang di isikan oleh guru.

7.5 Analisis

7.5.1 Analisis pengujian unit

Setelah melakukan pengujian unit yang menggunakan tiga sampel proses atau fungsi dari tiga *controller* yaitu proses login() dari *controller* C_index, proses tambah_kelas() dari *controller* C_guru, dan proses tukar_poin() dari *controller* C_Siswa. Untuk proses login() didapatkan nilai kompleksitasnya yaitu lima serta kasus uji yang dilakukan dengan mencoba lima jalur *independent path* yang didapat bernilai valid untuk semua jalur. Untuk proses tambah_kelas() didapatkan nilai kompleksitasnya yaitu tiga serta kasus uji yang dilakukan dengan mencoba tiga jalur *independent path* yang didapat bernilai valid untuk semua jalur. Untuk proses tukar_poin() didapatkan nilai kompleksitasnya yaitu lima serta kasus uji yang dilakukan dengan mencoba lima jalur *independent path* yang didapat bernilai valid untuk semua jalur. Dengan nilai kompleksitas terendah sama dengan tiga dan nilai kompleksitas tertinggi sama dengan lima maka sesuai dengan data yang ada pada Tabel 7.38, untuk ketiga sample tersebut termasuk memiliki kompleksitas yang rendah, dapat di uji dengan nilai rendah dengan artian sangat mudah untuk melakukan pengujian, dan juga memiliki tingkat *cost* yang rendah.

Tabel 7.38 Tingkatan angka *cyclometric complexity*

<i>Cyclometric Complexity</i>	Level Kompleksitas	Dapat Di Uji	Cost dan <i>Effort</i>
1-10	Rendah	Tinggi	Rendah
10-20	Tinggi	Sedang	Sedang
20-40	Sangat Tinggi	Rendah	Tinggi
>40	-	Tidak Dapat Di Uji	Sangat Tinggi

7.5.2 Analisis pengujian *requirement*

Data dari hasil pengujian *requirement* yang dilakukan pada semua fungsional yang telah tersedia pada aplikasi *e-learning* didapatkan hasil yang sangat baik yaitu semua kebutuhan fungsional telah *valid* ketika dilakukan uji dengan membuat prosedur uji pada tiap-tiap kebutuhan fungsional. Dengan hasil pengujian yang bernilai *valid* pada setiap kebutuhan fungsional berarti sistem ini dapat berjalan dengan baik dan sesuai dengan persyaratan pengguna.

7.5.3 Analisis *user engagement scale*

Berdasarkan hasil pengolahan data kuesioner *user engagement* didapatkan bahwa pengguna aplikasi *e-learning* merasakan hal yang sama ketika menggunakan aplikasi *E-learning*. Pengolahan data tersebut membuktikan bahwa tidak ada perbedaan signifikan ketika pengguna mengisi data kuesioner mereka yang dapat dilihat dari nilai standar deviasi yang memiliki nilai

dibawah satu. Ada 26 pertanyaan yang memiliki nilai standar deviasi dibawah satu, itu berarti menandakan jawaban para peserta tidak perbedaan yang signifikan.

7.5.4 Analisis *system usability scale*

Berdasarkan hasil perhitungan skor *system usability scale* didapatkan nilai 67,5. Nilai tersebut menggambarkan bahwa aplikasi *E-learning* mudah dipahami oleh pengguna dan dapat digunakan oleh pengguna. Nilai tersebut masuk kedalam kategori *ACCEPTABLE HIGH* yang berarti aplikasi *E-learning* bisa diterima oleh pengguna. Adanya keterbatasan pemahaman objek pengujian tentang teknologi informasi yang lebih khususnya pada sistem pembelajaran online dapat mempengaruhi skor dari *system usability scale*.



BAB 8 PENUTUP

8.1 Kesimpulan

Berdasarkan dari hasil analisis yang sudah dilakukan pada sub bab sebelumnya yaitu analisi, didapatkan kesimpulan sebagai berikut :

1. Hasil analisis kuesioner *user engagement scale* (UES) terdapat keseragaman antara tiga responden yang terlibat dalam pengujian aplikasi *e-learning*. Dari 30 pertanyaan yang diberikan kepada responden terdapat 26 pertanyaan yang memiliki standar deviasi dibawah satu yang menunjukkan tidak adanya perbedaan pendapat yang signifikan dari tiga responden tersebut. Dalam **Lampiran A** memaparkan pertanyaan yang sudah diberikan kepada responden. Tidak hanya menggunakan, penulis juga menggunakan *system usability scale* (SUS) untuk mendapatkan nilai kepuasan yang dapat mengacu kepada aplikasi *E-learning* dapat diterima oleh pengguna dalam artian mudah digunakan. Nilai SUS yang didapatkan adalah 67,5 dimana nilai tersebut termasuk kedalam rentan *ACCEPTABLE HIGH* yaitu antarmuka dapat diterima oleh pengguna. Untuk mendapatkan nilai tersebut peserta menggunakan data dari kuesioner SUS yang dilampirkan dalam **Lampiran B**. kurangnya pemahaman objek pengujian terhadap teknologi informasi dapat mempengaruhi nilai dari SUS, dengan adanya dua data diatas dapat disimpulkan bahwa pengguna tertarik dan puas dalam menggunakan aplikasi *E-learning* dengan menerapkan metode *gamification*.
2. Hasil perancangan yang telah dilakukan pada aplikasi *e-learning* ini menghasilkan beberapa *controller class*, *entity class*, *boundary class*, basis data, algoritme, dan antarmuka yang digunakan sebagai acuan untuk membangun atau membuat aplikasi *E-learning* dengan menerapkan metode gamifikasi. Sedangkan untuk hasil dari implementasi aplikasi *e-learning* ini yaitu menghasilkan fitur-fitur yang dapat membantu menjalankan aplikasi *e-learning* tersebut. Fitur-fitur yang dihasilkan dalam implementasi merupakan gambaran dari elemen-elemen yang ada dalam metode *gamification*.
3. Hasil dari pengujian menggunakan metode *whitebox testing* dan juga *blackbox testing*, tidak ditemukan adanya kesalahan pada aplikasi *e-learning*. Pengujian *whitebox* dilakukan dengan menggunakan pengujian unit, sedangkan *blackbox* dilakukan menggunakan pengujian *requirement* yang dilakukan pada seluruh kebutuhan fungsional yang ada. Dengan dilakukannya pengujian tersebut dapat membuktikan bahwa aplikasi *e-learning* dapat digunakan oleh pengguna dengan tidak adanya *error* atau kesalahan yang mengganggu.

8.2 Saran

Saran yang dapat diberikan penulis untuk membangun aplikasi *E-learning* dengan menerapkan metode *gamification* selanjutnya antara lain :

1. Fitur-fitur yang ada pada aplikasi *e-learning* lebih banyak menerapkan elemen-elemen *gamification*.
2. Perhitungan yang lebih mendalam terhadap pengujian yang menggunakan *user engagement scale* (UES), dan objek pengujiannya lebih besar atau banyak.



DAFTAR PUSTAKA

- Hu, P. J. H., & Hui, W., 2012. Examining the role of learning engagement in technology-mediated learning and its effects on learning effectiveness and satisfaction. *Decision Support Systems*, Volume 53, p. 782–792.
- F.A. Bachtiar, A. R. F. P., 2014. Acceptance in the deployment of blended learning as a learning resource in information technology and computer science program, Brawijaya University. *Computer Aided System Engineering (APCASE)*, pp. 131- 135.
- Lewis, C., 2002. Driving factors for e-Learning: an organizational perspective. *Perspectives*, 6(2), p. 50–54.
- Arbaugh, J. B. & D. R., 2002. Technological and structural characteristics, student learning and satisfaction with web-based courses – An exploratory study of two on-line MBA programs. *Management Learning*, 33(3), p. 331–347.
- Chen, W. L. C. & B. J. G., 2003. Understanding the dimensions of self-exploration in web-based learning environments. *Research on Technology in Education*, 34(3), p. 364–373.
- F. Pradana, F. B. B. P., 2018. PENGARUH ELEMEN GAMIFICATION TERHADAP HASIL BELAJAR SISWA PADA E-LEARNING PEMROGRAMAN JAVA. *Seminar Nasional Teknologi Informasi dan Multimedia*, pp. 5-7.
- Delon, W. & M. E., 1992. Information systems success. *The quest for the dependent variable. Information Systems Research*, 3(1), p. 60–95.
- Simões, J. R. R. D. & V. A. F., 2013. A social gamification framework for a K-6 learning platform. *Computers in Human Behavior*, Volume 29, p. 345–353.
- A. Amriani, A. F. A. A. .. Y. U. K. M. J., 2013. An Empirical Study of Gamification Impact on E-Learning Environment. *3rd International Conference on Computer Science and Network Technology*, pp. 265-269.
- K.M. Kapp, 2012. The Gamification of Learning and Instruction. *Pfeiffer*.
- F. L. Khaleel, N. S. T. S. M. T. W. A. I., 2016. Gamification Elements for Learning Applications. *International Journal on Advance Science Engineering Information Technology*, p. 872.
- Pressman, 2010. Software Engineering. In: *Software engineering : a practitioner's approach 7th ed*. Boston(Burr Ridge): Raghothaman Srinivasan, p. 1.
- Sommerville, I., 2003. *Software engineering : rekaya perangkat lunak jilid 1 edisi 6*. Jakarta: Erlangga.
- Sommerville, I., 2011. *Software engineering 9th ed*. London: Addison-Wesley.

- Ginanjari, 2017. Penerapan Metode Waterfall Pada Desain Sistem Informasi Geografis Industri Kabupaten Tegal. *Jurnal Informatika: Jurnal Pengembangan IT (JPIT)*, 2(1), p. 8.
- Sommerville, 2007. *Software Engineering (8th Edition)*. Harlow: Addison-Wesley.
- Rouibah, K. & A.-R. S., 2009. Requirement Engineering Elicitation Methods: A Kuwaiti Empirical Study About Familiarity, Usage and Perceived Value.. *Information Management & Computer Security*, pp. 192-217.
- Griffiths, 2010. *CodeIgniter 1.7 Professional Development*. BIRMINGHAM: PACKT.
- Blanco, 2009. *CodeIgniter 7.1*. Birmingham: s.n.
- Booch, 2007. *Object-Oriented Analysis and Design with Applications - 3rd Edition*. Boston: Pearson Education.
- Hendy, Yulistia, 2013. Sistem Informasi Pemesanan Dan Penjualan Berbasis Web Pada Dewi Florist. *Sistem Informasi*, p. 3.
- Alawiyah, 2017. *Pengembangan Aplikasi E-SHCOOL Dengan Pendekatan Evolutionary Prototype Studi Kasus SMP Negeri 1 Cikarang Barat*. Malang: Ruang Baca Filkom UB.
- Madhavi, 2016. A White Box Testing Techniques in Software Testing : Basis Path Testing. *Journal for Research*, 02(04), pp. 12-17.
- Masrom, M., 2017. Technology Acceptance Model and E-Learning. *12th International Conference on Education, Sultan Hassanali Bolkiah Institute of Education, University Brunei Darussalam 21-24 May 2007*.
- Davis, F., 1989. Perceived usefulness, perceived ease of use and user acceptance of information technology. *MIS Quarterly*, 13(3), pp. 319-339.
- Hershey, 2012. *SDLC: Importance of Requirement Analysis*. [Online] Available at: <http://www.codespread.com/sdlc-importance-of-requirement-analysis.html> [Accessed Desember 2016].
- E. Priyambodo, A. W. R. L. P. S., 2012. PENGARUH MEDIA PEMBELAJARAN INTERAKTIF BERBASIS WEB TERHADAP MOTIVASI BELAJAR MAHASISWA. *Jurnal Kependidikan*, 11, 42(2), pp. 99-109.
- M. Urh, G. V. E. J. R. P., 2015. The model for introduction of gamification into e-learning in higher education. *Procedia - Social and Behavioral Sciences* 197, p. 388 – 397.
- Turkoguz, S., 2012. Learn to Teach Chemistry Using Visual Media Tools. *Chem. Educ. Res. Pract.* 13 July 2012, DOI: 10.1039/c2rp20046e.
- Lee, J. & H. J., 2011. Gamification in education: What, how, why bother?. *Academic Exchange Quarterly*, 15(2).
- Flowler, M., 2003. *UML Distilled: A Brief Guide to the Standard Object Modeling Language (3rd Edition)*. Boston: Addison Wesley.

- Kurniawan, T. A., 2018. PEMODELAN USE CASE (UML): EVALUASI TERHADAP BEBERAPA KESALAHAN DALAM PRAKTIK. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, 5(1), pp. 77-86.
- Larman, C., 2004. Applying UML and Patterns : An Introduction to Object-Oriented Analysis and Design and Iterative Development. *3rd Edition*, October.
- Cockburn, A., 2000. Writing Effective Use Cases. 1st Edition. In: *Humans and Technology*. Boston(MA): Addison- Wesley Longman Publishing Co., Inc..
- Heather L. O'Brien, P. C. M. H., 2018. A practical approach to measuring user engagement with the refined user engagement scale (UES) and new UES short form. *International Journal of Human-Computer Studies*, Volume 112, pp. 28-39.
- Brooke, J., 1986. *SUS - A quick and dirty usability scale*. [Online] Available at: <https://hell.meiert.org/core/pdf/sus.pdf> [Accessed 13 10 2018].
- Ardiansyah, M. I. G., 2016. PENGUJIAN USABILITY USER INTERFACE DAN USER EXPERIENCE APLIKASI E-READER SKRIPSI BERBASIS HYPERTEXT. *Jurnal Ilmiah Teknologi Informasi Terapan*, II(13), pp. 213-220.

